Computational methods for investigating cell motility with applications to neutrophil cell migration

Konstantinos N. Blazakis

Submitted for the degree of Doctor of Philosophy

University of Sussex

January 2015

Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Konstantinos N. Blazakis

Dedication

To Athina and Nikos.

Acknowledgements

The period while I was working on this Ph.D. thesis has been a wonderful, productive and often overwhelming experience. It would not have become possible to finish without the help and support of extraordinary people around me.

First and foremost, I am deeply grateful to my supervisors Dr Anotida Madzvamuse and Dr Vanessa Styles for their support, depth of knowledge and patience throughout this thesis. I would also like to thank Dr Chandrasekhar Venkataraman for his assistance and co-supervision. I have been lucky that they have introduced me into this research and provided me with skills necessary to write this thesis. Without them this thesis would not have been possible. In addition, I would like to thank Dr Constantino Carlos Reyes-Aldasoro from City University-London for providing the necessary experimental data for this thesis and for an alternative interpretation of my numerical outcomes. His "engineering" vision provided helpful and alternative perspectives on my work. It is an honour to work with such a world leading research personalities and kind persons.

On a practical level, I wish to express my gratitude to "Ebirikion Foundation Grant" in Greece for funding partially my studies and the High Performance Cluster at the University of Sussex for providing the necessary computational facilities for my work. In addition, I wish to thank the Doctoral School from the University of Sussex for various travel and conference grants and the MASAMU project (a tri-collaborative effort between the University of Sussex, Auburn University (USA) and the Southern Mathematical Sciences Association (SAMSA) in Africa) for an ultimate experience that allowed me to interact with other pioneering researchers in mathematical science from the Southern Africa and the USA.

Many thanks to Dr Muhammad Abdullahi Yau, Dr Haidar Haidar, Miss Elham Khairi, Mr Andy Chung, Mr Giannis Neofytou, Mr Hussaini Ndakwo, Mrs Rattana

Prapanporn and Mrs Laurrie Trot who helped to create a nice and warm working atmosphere during my Ph.D. It was great to have such nice people around me. Also, I would like to thank the support staff of the school office and Mrs Liz Davis from IT Services, for their constant support throughout my study. Special thanks, also, to Mr Tom Armour for making available all the software and computing tools that were necessary for this work.

I would also like to thank my loyal friends back home in Greece. In addition, many thanks to Sophia, Filippos, Poppy and the new lifelong friend Alexia, who have made my time in the UK such an enjoyable and memorable experience. I would like to express the final and most important thanks to my housemate and lifetime friend Thodoris. It is fair to say that I would not have been able to get anywhere and fulfil this thesis without his precious help.

My deepest debt of gratitude to my father Nikos, my sister Antriana and to the memory of my mother Athina, for their enormous support and advice throughout my life. Finally, I would like to thank the "new" arrivals in my family Eftihis and Dimitris, who were part of my thoughts during the last stages of this work.

Computational methods for investigating cell motility with applications to neutrophil cell migration

Konstantinos N. Blazakis

Submitted for the degree of Doctor of Philosophy

University of Sussex, January 2015

Abstract

Cell motility is closely linked to many important physiological and pathological events such as the immune response, wound healing, tissue differentiation, embryogenesis, inflammation, tumour invasion and metastasis. Understanding the ability of cells to alter their shape, deform and migrate is of vital importance in many biological studies.

The rapid development in microscopy and imaging techniques has generated a huge amount of discrete data on migrating cells *in vivo* and *in vitro*. A key challenge is the use of discrete experimental observations to develop novel methods and algorithms that track cells and construct continuous trajectories of their motion as well as characterising key geometric quantities associated with cell migration. Therefore, in this work using robust numerical tools we focus on proposing and implementing mathematical methodologies for cell movement and apply them to model neutrophil cell migration.

We derive and implement a computational framework that encompasses modelling of cell motility and cell tracking based on phase field and optimal control theory. The cell membrane is represented by an evolving curve and approximated by a diffuse interface; while the motion of the cell is driven by a force balance acting normal on the cell membrane. This approach allows us to characterise the locus of the centroid cell-surface position.

In addition, we describe a surface partial differential equation framework that can be coupled with the phase-field framework, thereby offering a wholistic approach for modelling biochemical processes and biomechanics properties associated with cell migration.

Publications from this thesis

Here we detail the following papers that have been published from this thesis:

- Chapter 4 is part of the following paper (Blazakis et al. [2015b]):

  Blazakis, K., Madzvamuse, A., Reyes-Aldasoro, C. C., Styles, V. and Venkataraman, C. (2015) *Whole cell tracking through the optimal control of geometric evolution laws.* Journal of Computational Physics, 297. pp. 495-514. ISSN 0021-9991

- Chapter 5 is part of the following paper (Blazakis et al. [2015a]):

  Blazakis, K., Reyes-Aldasoro, C. C., Styles, V., Venkataraman, C. and Madzvamuse, A. (2015) *An optimal control approach to cell tracking.* In: Louis, Alfred K, Arridge, Simon and Rundell, Bill (eds.) Proceedings of the Inverse Problems from Theory to Applications Conference (IPTA2014). IOP Publishing Ltd , pp. 74-77. ISBN 9780750311069

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General overview

Cell migration is a fundamental process in cell biology and is tightly linked to many important physiological and pathological events (Le Clainche and Carlier [2008]) such as the immune response (Ridley et al. [2003]), wound healing (Kadirkamanathan et al. [2012]), tissue differentiation (Xue et al. [2010]), metastasis (Wolf et al. [2003]; Fletcher and Theriot [2004]), embryogenesis, inflammation and tumour invasion (Bray [2001]). Consequently, failure of cells to migrate, or inappropriate migratory movements, can result in severe defects or life-threatening scenarios such as defective wound repair, tumour dissemination or autoimmune diseases (Horwitz and Webb [2003]). Thus, there is an imperative need of studying the mechanisms that are linked with these processes.

In addition, experimental advances provide techniques to observe migrating cells both *in vivo* and *in vitro*. Inferring dynamic quantities from this static data is an important and challenging task that has many applications in unravelling the mechanisms related to cell motility. On the other hand, a major focus of current research is the derivation of mathematical models for cell migration based on physical principles (Neilson et al. [2010]; Ziebert et al. [2011]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]; Shao et al. [2012]; Marth and Voigt [2013]). Croft et al. [2013] investigated fitting parameters in models for cell motility to experimental image data sets of migrating cells. The available observations included both the positions of the cells and the concentrations of cell-resident proteins, which are related

to cell motility. Despite this, very little research has been focused on incorporating these mathematical modelling advances into appropriate cell tracking algorithms.

Cell tracking refers to the non-invasive imaging of cells, that have been labelled, with the aid of fluorescence *in vivo* microscopy, in an effort to monitor their location and analyse their behaviour (Maška et al. [2014]). The study of cell migration and morphology is one of the main areas for biomedical research as through their understanding it may lead to prevention or even the cure of diseases (Zimmer et al. [2006]). The main difficulty in that effort emanates from the vast available data as well as due to the nature of the cells or particles which appear in large numbers and vary in density (Meijering et al. [2012]). Thus, computational tools are necessary in order to efficiently analyse the humongous wealth of experimental data available. An excellent example is the better interpretation of chemotaxis due to advance image processing algorithms that permit the pseudopod tracking and detection (Xiong and Iglesias [2010]; Bosgraaf and Van Haastert [2010]). Similar computerised methods are already being used for cell tracking requiring the synergy of robust algorithms. We refer the reader to Zimmer et al. [2006], Meijering et al. [2009], Hand et al. [2009], Xiong and Iglesias [2010] and Meijering et al. [2012] for a literature review comparing and presenting the main features of the most commonly software packages for cell tracking that are being used by experimentalists.

In general, cell tracking consists of three main steps (Meijering et al. [2012]):

1. *Segmentation*: In order to track the cells from static imaging data, the cells must be first determined and then located. The process during which the cells in an image are separated from the background is known as segmentation (Meijering et al. [2009]). The most commonly used methods for cell segmentation include thresholding, template matching and deformable models among others (Miura [2005]; Hand et al. [2009]; Meijering et al. [2009]; Xiong and Iglesias [2010]; Meijering et al. [2012]). In "thresholding" by setting an intensity threshold value, the image is separated into two regions; regions above this intensity threshold value are considered as foreground, whereas those below the predefined value are classified as background (Hand et al. [2009]; Meijering et al. [2009]). Another type of "thresholding" is the "hysteresis" in which a dual "thresholding" operation is performed on an image using two threshold

values (lower and upper) (Henry et al. [2013b]). The "template matching" method fits predetermined intensity templates to the image data. However, this technique does not work in cases where the cell-shape changes over the time (Meijering et al. [2009]; Meijering et al. [2012]). Finally, the use of "deformable models" for cell segmentation has gathered, in recent years a huge amount of interest from theoreticians and experimentalists. The cell membrane is defined either explicitly as a parametric contour model or implicitly as the zero level set of a scalar function. The latter provides a more powerful approach that allows for topological changes such as cell division. After the initialisation of the contour around the object of interest, deformable models are evolved to minimise a predefined energy functional. This energy functional incorporates terms related to the image (such as intensity) and the segmented object (such as shape's curvature and/or surface area) (Miura [2005]; Zimmer et al. [2006]; Meijering et al. [2009]; Tyson et al. [2010]). We refer to Miura [2005], Hand et al. [2009] and Meijering et al. [2009] for a review and comparison of the most prevalent methods for cell segmentation.

2. *Matching and Linking*: The cells segmented in the first step must then be associated from frame to frame (note this is only relevant in the case of multiple cell image data sets) such that where possible (in practice cells may disappear or spontaneously appear in images) there is a one-to-one map that uniquely associates individual cells from one frame to the next. The simplest and frequently used approach is the technique that identifies the spatially "nearest" cell in the next frame (Meijering et al. [2009]; Meijering et al. [2012]). In most of the cases the spatial distance measures the distance between the cell centroid positions in the successive frames. However, sometimes the "nearest" may refer to other shape features, apart from the spatial distance. These features could include the area, orientation, perimeter or the cell's boundary curvature (Meijering et al. [2009]; Meijering et al. [2012]). Another technique besides the "nearest" is the one which predicts the next position using areas of probability (Reyes-Aldasoro et al. [2008]). Alternatively, some of the techniques employed for the cell segmentation solve the linking problem implicitly. We refer to Miura [2005] and Hand et al. [2009] for more details related to cell

association from one frame to the next.

3. *Tracking measures*: The direct result of applying tracking methodologies is a sequence of coordinates, indicating the locomotion of the cell at every time step. Certainly, one of the most exciting outcome is the computation of biologically meaningful quantities derived from the associated segmented static cells (Meijering et al. [2012]). These dynamic data include measures related to the motility and the morphology of the moving cells. Parameters that are associated with the quantification of the cell migration behaviour consist of the construction of the cell trajectory (mostly the cell centroid), the persistence length, the chemotactic index, the mean squared displacement and the cell speed (magnitude of the velocity) (Beltman et al. [2009]; Meijering et al. [2012]; Cheng et al. [2007]). Furthermore, quantities, among others, that measure the size and the complexity of the cell shape include volume, surface area and sphericity (Meijering et al. [2012]; Reyes-Aldasoro et al. [2009]).

1.2 Objective of this study

The objective of this thesis is to serve as a useful first step in the development of whole cell tracking algorithms which track the morphology of whole the cell rather than particle tracking in which particles such as the cell centroid or cell resident proteins or (macro-)molecules are tracked (Henry et al. [2013b]). The underlying model for the cell evolution in the proposed algorithms, is based on physical principles, rather than purely geometric considerations, e.g. equidistribution of vertices (Dormann et al. [2002]; Bosgraaf et al. [2009]). In this setting, one hopes to attain estimates of motility-related features such as trajectories, velocities, confinement ratios, etc., which reflect the physics underlying the model.

We stress the fact that the tracking procedure we propose allows us to incorporate physically important aspects of cell migration by including, for example, volume conservation in the model for the cell evolution. This is motivated by the observation that, for many cells, while the surface area of the cell membrane may change significantly during migration the volume enclosed by the cell remains roughly constant (Shao et al. [2010]).

One feature of the existing cell tracking methods is that the trajectories they generate are not physical in nature, they are designed rather with the goal of achieving nice geometric properties. Our approach differs in that we start with a model derived from physical principles and it is this model for the evolution that drives the tracking algorithm. In this sense the method that we will propose later in this thesis, is similar in spirit to the parameter identification procedure described in Croft et al. [2013] as in both studies the goal may be regarded as fitting a mathematical model to experimental image data sets.

In this thesis we will largely neglect the segmentation step which is associated with the majority of the cell tracking algorithms. We assume either that we have segmented image data to work with or that the image data is of sufficient quality that the contrast between the cell and the background is clear and a simple thresholding step is sufficient to label the different cells. In the case of segmented image data, we assume this data consists of closed surfaces (or curves in $2d$) that describe the boundaries of each individual cell.

The novelty and significance of the proposed algorithm lies in the fact that it is, to our best knowledge, the first study that seeks to bring together research at the forefront of modelling cell migration with cell tracking. This collaboration uses aspects associated with the bio-physics of the cells, such as volume conservation and the resistance of the cell boundary to stretching, and the location of the cell throughout the movement. We derive and implement a computational framework that encompasses modelling of cell motility and cell tracking based on phase field and optimal control theory. The evolution of the cell membrane is modelled as being governed by volume conserved mean curvature flow with forcing (we refer to (4.5) for more details).

This evolution law is a simplification of a large class of models that arise in the modelling of cell motility (Neilson et al. [2010]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). The forcing function that enters the evolution law is the main driver of the directed migration and serving as a control variable for the cell-surface evolution. The model we present is phenomenological and hence it is difficult to directly relate this control function to biophysical processes. However, our numerical findings indicate that one interpretation of this forcing function is

that it accounts for both protrusive and contractile forces which are associated with the cell movement.

As we mentioned earlier we formulate this problem within the phase-field framework. The phase-field method is a powerful computational method that can handle complex topological changes. Phase-field modelling has been used in a broad range of applications in material science (Steinbach [2009]), fluid dynamics (Anderson et al. [1998]), biology (Du et al. [2005]) and in solving phase transition problems. In addition, the phase-field method has been applied to a wide range of applications related to cell motility (e.g., Ziebert et al. [2011]; Marth and Voigt [2013]). Within this phase-field formulation, the moving cell membrane $\Gamma(t)$ is represented by an evolving thin interfacial layer involving a small parameter ε related to its thickness (we refer to the introduction of Chapter 4 for more details). The evolution of the cell membrane can be approximated by a phase-field model which is a driven Allen-Cahn (Evans et al. [1992]) equation with forcing.

Then, we formulate the cell tracking problem of fitting a mathematical model for cell motility to the experimental data. This problem can be considered as a partial differential equation constrained optimal control problem. We define the objective functional (4.4) which has to be minimised for a space and time control function. The goal is to drive the cell-surface using the control function, in such way that the position of the cell membrane at the final time $t = T$ is close as possible to the desired given position. By adopting an optimal control approach (see §4.2.3) for the solution of the minimisation problem, we formally derive the first order necessary conditions ([Tröltzsch, 2010]). The system of the state and adjoint equations is solved by using the finite element method (Deckelnick et al. [2005]), and the control variable is computed by a gradient update scheme (we refer to §4.3 for more details regarding the derivation of the state and adjoint equations which are involved in this problem and the gradient update scheme).

This study has the intention to recover physically meaningful dynamic data from static imaging data sets. We hope that this algorithm will spur future studies into incorporating advances in the modelling of cell migration into cell tracking algorithms. Such algorithms could be considered as an ad-hoc approach to cell tracking for recovering trajectories and velocities from discrete image data for both

experimentalists and theoreticians.

In addition to the above study, this thesis will present a candidate numerical method to tackle problems that will encompass the modelling of cell motility and cell tracking. By defining the surface geometry of migrating cells from the cell tracking algorithm at each time step, we can propose innovative mathematical models for cell motility. Here, we will introduce the surface finite element method (Dziuk and Elliott [2007b]), that may allow us to solve efficiently and robustly such problems. We will apply this methodology to solve partial differential equations, specifically those of reaction-diffusion type posed on stationary surfaces.

Such equations have been used widely at cell motility studies and take into account several interacting chemical species, that are associated with the cell motion and live on the cell membrane. In addition, their solutions have been used to drive mechanical models of the protrusive and retractive forces exerted on the cell boundary (e.g., Neilson et al. [2010]; Ziebert et al. [2011]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). Moreover, these solutions can be coupled within the evolution law that it has been used to drive the cell tracking algorithm. They can be considered as an additional force that it is associated with the concentrations of bulk and/or surface resident chemical species.

1.3 Materials

The use of animal models represent an important tool for the biomedical researchers in order to investigate the pathological procedures of human diseases and to promote new therapeutic strategies. An example of this collaboration is the zebrafish *Danio rerio* (Lieschke and Currie [2007]). The zebrafish known as *Danio rerio*, is a small tropical freshwater fish, named for the five uniform, pigmented, horizontal blue stripes on the side of its body, all of which extend until the end of its caudal fin (see Figure 1.1) (Ganguly et al. [2012]; Spence et al. [2008]).

Neutrophil migration as observed in zebrafish larvae has become a popular model to study cellular inflammatory response. This is mainly due to the transparent nature of the zebrafish in the developmental stage and the capacity of genetic manipulation which allows the design of experiments to reproduce *in vivo* conditions of

inflammation similar to those that humans may experience (Renshaw et al. [2006]; Henry et al. [2013a]). Regarding the former, with the use of green fluorescent protein (GFP) experimentalists are able to label neutrophils and track their movements by video microscopy (Kadirkamanathan et al. [2012]).

Figure 1.1: Adult zebrafish *Danio rerio*. Figure is taken from the website `http://en.wikipedia.org/wiki/Zebrafish`.

Within the immune system neutrophils, both in humans and animals, play a vital role as they represent a daily ally against infectious diseases (Reyes-Aldasoro et al. [2009]). Compared to other migrating cells, one of their main attribute is the fast response to regions where there is an infection or tissue injury (Elks et al. [2011]; Holmes et al. [2012]). In addition, neutrophils have the ability to sense and detect external chemical messengers called "chemoattractants", and guided by their chemical cue they migrate towards the area in need (Nathan [2006]). Once neutrophils come in contact with pathogens they become "activated". One early sign of this "activation" could be considered the deformation of their shape, though this relation is still an open research topic (Reyes-Aldasoro et al. [2009]). As a result, the importance of understanding neutrophil's movement and behaviour for the defence against disease, as well as for the possible harmful outcome in cases of improper activation becomes obvious (Edwards [2005]; Reyes-Aldasoro et al. [2009]).

In this thesis we are using biological data from neutrophils as observed in zebrafish *Danio rerio larvae*. In particular, a tail fin transection (see Figure 1.2) was performed at 3 days post fertilisation and images were captured using an UltraVIEW-VoX spinning disk confocal microscope (PerkinElmer Life and Analytical Sciences) (Renshaw et al. [2006]). Following this transection an inflammatory response was elicited, evoking the migration of neutrophils to the site of injury. We selected eight

observations of the migration of a single neutrophil and we are taking cross-sections of the three-dimensional data, as shown in Figure 1.3. The orientation is such that tail fin is to the right of the neutrophils. This dataset of neutrophils will be used to

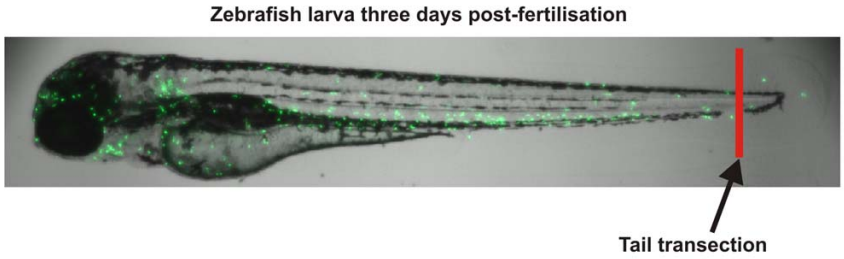


Figure 1.2: Neutrophils as observed in zebrafish *Danio rerio larvae* are visible by using green fluorescent protein (GFP). The picture is taken from Kadirkamanathan et al. [2012].

address the challenging task of validating the proposed cell tracking algorithm. We will present a mathematical approach for cell tracking, in which we formulate the neutrophil tracking problem as an inverse problem for fitting a mathematical model for cell motility to this experimental imaging data. Once the continuous trajectory of the migrating neutrophil is established, quantitative measures that are related to the motility and the morphology of the neutrophils will be extracted.

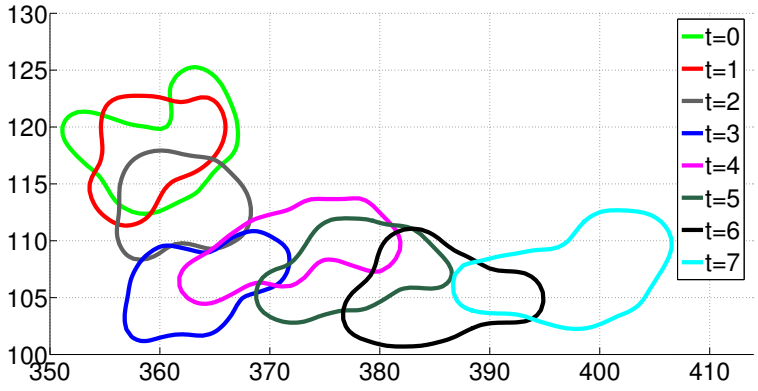


Figure 1.3: Experimental data representing a discrete set of the locations of the cell membrane corresponding to the migration of a single neutrophil as observed in zebrafish *Danio rerio larvae* (Henry et al. [2013b]).

1.4 Outline of the thesis

Below we outline the structure of our thesis. In Chapter 2 we formulate and solve partial differential equations of reaction-diffusion type on stationary surfaces using the surface finite element method (Dziuk and Elliott [2007b]). Here, we are setting foundations to couple in the future the cell tracking algorithm to reaction-diffusion models posed on the cell surface that describe the spatiotemporal behaviour of chemical species resident on the cell surface. We start by stating preliminaries and introducing the notation for the surface calculus that will be used throughout this chapter. Furthermore, in order to demonstrate the applicability and the generality of the surface finite element method, we present numerical simulations on two different surfaces: a sphere of radius one and the surface of a neutrophil from the zebrafish *Danio rerio larvae*.

In Chapter 3 we employ an interpolation method in order to propose a simple cell tracking algorithm. By using cubic splines, a series of intermediate cell boundaries are generated that describe and track the cell evolution. Although there is no physical evolution law associated with cubic interpolation, nevertheless it is one way of obtaining continuous trajectories of the cell.

In Chapter 4 we present a mathematical approach for cell tracking, where only the position of the cell, specifically the cell membrane, at a series of discrete times is available and no further biological information is given. We derive and implement a computational framework that encompasses modelling of cell motility and cell tracking based on phase field and optimal control theory. The cell membrane is represented by an evolving curve and approximated by a diffuse interface; while the motion of the cell is driven by a force balance equation. Thus, controlling the evolution of the interface becomes a standard distributed control problem.

The novelty of the method is that the tracking algorithm is driven by a model for the motion of the cell, based on physical principles for cell migration. Then we formulate and solve an inverse problem, which takes the form of a partial differential equation constrained optimisation problem, for fitting the model to the experimental observations.

Finally, we present numerical results that validate our methodology. In order to recover the whole cell morphologies through time, we implement our algorithm using

synthetic biological data. In addition, we apply the proposed algorithm on multiple synthetic cell data and discuss some attractive features of our methodology that are associated with the segmentation and matching problem. The chapter finishes by discussing some of the limitations of our algorithm and proposing future research directions.

Chapter 5 contains an application of the optimal control algorithm to experimental data on the migration of neutrophils as observed in zebrafish *Danio rerio larvae*. In order to be able to contrast this methodology, we compare the numerical results with those obtained in Chapter 3, based on cubic interpolation.

The thesis is completed in Chapter 6, where we summarise the main results of our work. It is in this chapter that we discuss the implications of our work. Furthermore, we discuss possible research extensions associated with the cell tracking.

1.5 Technical details

The implementation of the numerical schemes in Chapters 2 and 4 were carried out by using the package of ALBERTA-2.0 (Schmidt and Siebert [2005]). This is a flexible and efficient finite element toolbox written in C programming language. Its main feature is the availability of data structures for easy assembly of a discretised problem. In addition, all the graphics in these chapters were produced using the visualisation application ParaView (Henderson et al. [2004]).

Matlab version 7.13.0 (R2011b) has been employed for the implementation of the computational methods that we executed in Chapter 3. Simulations were performed on a Linux machine with operating system CentOS, with an Intel Xeon at 1.9 GHz and 2Gb of RAM and on MacBook pro (OS X version 10.9.3), with 2.9 GHz Intel Core i7 and 8GB RAM.

All the numerical experiments in Chapters 4 and 5 have been performed on the high performance cluster (HPC) at the University of Sussex. Each of the simulations was carried out in serial using a single core of the cluster. The HPC cluster currently consists of 3140 cores with an even mixture of Intel and AMD CPUs. The majority of the cluster are 64 core AMD nodes with 256GB RAM per node, and a smaller number of 512GB RAM nodes. The cluster uses the high-performance Lustre clustered-

filesystem for I/O, and currently stands at 298TB of storage for research use.

Chapter 2

Surface Finite Element Method

2.1 Introduction

In this chapter we introduce the surface finite element method that we will employ to solve partial differential equations, specifically those of reaction-diffusion type posed on stationary surfaces. Surface partial differential equations are employed to model a wide range of applications in fluid mechanics, image processing, cell biology and material science (Turk [1991]; Tang et al. [2005]; Lefèvre and Mangin [2010]). For example, in many applications in cell biology it is of vital importance for the integrated understanding of many cellular biology processes to couple the internal (bulk) and surface dynamics (Heys et al. [2006]). Furthermore, in many studies for the modelling of cell motility and chemotaxis, surface partial differential equations on the cell boundary have been used to describe the biochemistry of the models account for gradient sensing and polarisation (Neilson et al. [2010]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). The motion of the cell boundary is described by a geometric evolution equation. Thus, by coupling the biochemistry and the mechanics, robust computational frameworks have been developed for investigating cell motility.

Although, this thesis is devoted to computational methods for investigating cell motility, we have focused on the case in which no information regarding the biochemistry of the examined cells is given. In the following chapters, we will describe a mathematical framework for the cell tracking based on an evolution law for the cell membrane, that has been recently proposed in the research of cell motility (Neilson

et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). This approach can be applied and generalised to models including more biological aspects. In particular models for the dynamics of actin and myosin, as well as other motility related species within the cell and on the membrane may be included. The evolution law may be modified to take into account the dependence of the movement of the cell on these species. In such models partial differential equations on the cell membrane are necessary. Motivated by this, the aim of this chapter is to provide the foundation for tackling such problems.

Various numerical methods have been proposed for solving partial differential equations both on stationary and evolving surfaces. The choice of the numerical method is strongly linked to the different ways of representing the surface. To this end, we outline next the main approaches of describing a surface (Deckelnick et al. [2005]).

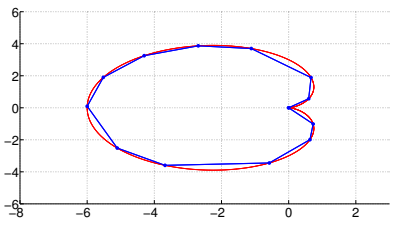
Triangulated surfaces. A surface is approximated by a polygonal surface whose vertices lie on the surface and the calculations are performed on this discrete representation. It is clear that refining this polyhedral, the triangulated surface tends to match up with the surface (see Figure 2.1(a)).

Level set methods. With the level set methods, a surface can be represented implicitly, where we consider the surface as the zero level set of an auxiliary function (Sethian [1999]; Fedkiw [2003]). One of the main advantages of the level set representation of a surface is that it is possible to capture topological changes, especially when the surface is evolving (see Figure 2.1 (b)).

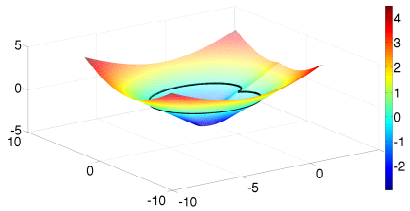
Phase field approach. With the phase field representation we approximate the surface interface by a "diffuse surface" of width $\mathcal{O}(\varepsilon)$ and the zero level set of the phase field function $\varphi_\varepsilon(\boldsymbol{x}, t) : \Omega \times (0, T) \to \mathbb{R}$, approximates the surface

$$\Gamma_\varepsilon(t) = \{\boldsymbol{x} \in \mathbb{R}^{n+1} | -1 + c_\varepsilon \leq \varphi_\varepsilon(\mathbf{x}, t) \leq 1 - c_\varepsilon\},$$

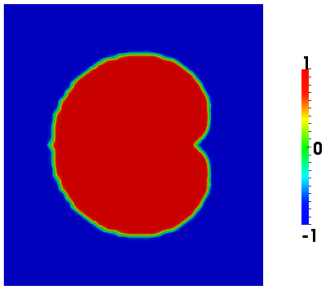
where Ω is a bounded domain in $\mathbb{R}^{n+1}$. The phase field function $\varphi(\boldsymbol{x}, t)$ takes values from -1 to $+1$ (Deckelnick et al. [2005]). This approach is another example of an implicit representation of a surface and has the capability to track topological changes that occur while a surface is evolving (see Figure 2.1 (c)). The bulk values of the phase field function correspond to the minima of the double well potential.



(a) Triangulated surface



(b) Level set representation



(c) Phase field representation

Figure 2.1: Description of a curve $\Gamma = \{x \in \mathbb{R}^2 | (x_1^2 + x_2^2 + 3x_1^2)^2 - 3^2(x_1^2 + x_2^2) = 0\}$. (a) The blue line is a triangulated surface approximation of the curve Γ. (b) Level set representation of the curve Γ. The black line is the zero level set. (c) A phase field representation of the curve Γ with $\varepsilon = 0.1$.

We refer to Chapter 4 for more details about the the phase field approach.

Based on the above different descriptions of the surfaces several numerical methods have been designed for approximating the solution of partial differential equations on surfaces. The most commonly used numerical methods for the solution of partial differential equations on surfaces are: the surface finite element method on triangulated surfaces (Dziuk [1988]; Deckelnick et al. [2005]; Dziuk and Elliott [2007a]; Dziuk and Elliott [2007b]; Barreira et al. [2011]; Dziuk and Elliott [2013]), the implicit finite element method where the surface is approximated with a level set representation (Fedkiw [2003]; Dziuk and Elliott [2008]; Burger [2009]; Dziuk and Elliott [2010]), the use of phase field methods (Deckelnick et al. [2005]; Deckelnick et al. [2001]; Rätz and Voigt [2006]; Elliott et al. [2010]), particle methods based on the level set representation of the surfaces (Hieber and Koumoutsakos [2005]; Bergdorf et al. [2010]), and the closest-point methods (Ruuth and Merriman [2008];

Macdonald et al. [2013]).

As we mentioned earlier partial differential equations posed on surfaces arise in a variety of applications such as in developmental biology, where morphogenesis constitutes one of its main aspects. Morphogenesis is a phenomenon which involves the development and formation of pattern and shape in biology and it is governed by many processes (Murray [2002]). Thus, understanding the mechanisms underlining the emergence of patterns and shapes is an important task for theoreticians and experimentalists alike. In 1952, Alan Turing proposed that the process of morphogenesis is a simple mechanism where a reaction between chemical species (morphogens) and a diffusion through tissues occur (Turing [1952]). Turing showed that systems which are stable in the absence of diffusion could become unstable in the presence of diffusion, thereby giving rise to the formation of spatial structures.

This pioneering work of Turing provoked a continuous research effort, concerning the modelling of the emergence of spatial structures during growth development in many fields such as ecology, chemistry, developmental biology and physics. Recent studies have focused on encompassing surface features such as the surface growth, shape and curvature into the modelling (Crampin et al. [1999]; Madzvamuse [2000]; Plaza et al. [2004]; Barreira et al. [2011]). However, in this chapter we are only considering stationary surfaces, our numerical simulations conclude that the surface geometry and the curvature are key factors in the emergence of patterns. Also, we remark that the surface finite element method has been extended to track problems on evolving surfaces by Dziuk and Elliott [2007a] with further analysis given by Dziuk and Elliott [2012], Lubich et al. [2013] and Dziuk and Elliott [2013].

We start this chapter by presenting in Section 2.2 definitions and theorems that are important in order to establish the method that will follow. Then, in Section 2.3 we describe the standard derivation of reaction-diffusion equations on stationary surfaces. In Section 2.4 we introduce the model of reaction kinetics that is used for all the undertaken numerical simulations. The details of the surface finite element method are presented in Section 2.5. Finally, in Section 2.6 we illustrate the effectiveness and applicability of the surface finite element method, performing several numerical simulations on arbitrary surfaces. For illustrative purposes, we consider two examples. In the first we take the smooth surface of a sphere with radius one for

our computations. The second example is inspired by experimental observations of neutrophils from the zebrafish *Danio rerio larvae* during the process of wound healing. The chapter finishes with a discussion of our numerical findings and suggestions for future work.

2.2 Definitions and formulae

In this section we recall some basic notation and important theorems that are used throughout this thesis. Also, we introduce some essential definitions and we establish the notation used later in this chapter. We refer to Dziuk and Elliott [2013]; Dziuk and Elliott [2007a]; Dziuk and Elliott [2007b]; Deckelnick et al. [2005] and the references therein, for more details.

We start by summarising appropriate function spaces and their corresponding norms. Let Ω be an open set in $\mathbb{R}^n$ and ξ a continuous real-valued function defined on Ω. We define by $L_p(\Omega)$ the class of the Lebesgue integrable functions defined on Ω such that

$$L_p(\Omega) := \left\{ \xi : \int_\Omega |\xi(x)|^p dx < +\infty \right\},$$

where p is a real number, $p \geq 1$. $L_p(\Omega)$ is equipped with the norm

$$||\xi||^p_{L_p(\Omega)} := \int_\Omega |\xi(x)|^p \, \mathrm{d}x.$$

When $p = +\infty$ we define the space $L_\infty(\Omega) := \{\xi : \operatorname{ess\,sup}_{x\in\Omega} |\xi(x)| < +\infty\}$ and the corresponding norm $||\xi||_{L_\infty(\Omega)} := \operatorname{ess\,sup}_{x\in\Omega} |\xi(x)|$. By

$$W^k_p(\Omega) = \left\{ \xi \in L_p(\Omega) : D^\alpha \xi \in L_p(\Omega), \quad |a| \leq k \right\}$$

we define the Sobolev space of order k; where k is a non-negative integer and by

$$D^\alpha \xi = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1} \dots \left(\frac{\partial}{\partial x_n}\right)^{\alpha_n} \xi$$

we denote a multi index notation for partial derivatives (in weak sense) and $|\alpha| := \sum_{i=1}^n \alpha_i$ is the length of α. The Sobolev norm is defined by

$$||\xi||_{W^k_p(\Omega)} := \begin{cases} \left(\sum_{|a|\leq k} ||D^\alpha \xi||^p_{L_p(\Omega)} \right)^{\frac{1}{p}}, & \text{when } 1 \leq p < \infty, \\ \\ \sum_{|a|\leq k} ||D^\alpha \xi||_{L_\infty(\Omega)}, & \text{when } p = \infty. \end{cases}$$

Finally, the Hilbert space is defined by

$$H^k(\Omega) := W_2^k(\Omega)$$

and its norm

$$||\xi||^2_{H^k(\Omega)} := ||\xi||^2_{L_2(\Omega)} + \sum_{i=1}^{k} |\xi|^2_{H^1(\Omega)},$$

respectively. With $|\xi|_{H^1(\Omega)} := ||D^k\xi||_{L_2(\Omega)}$ we denote the seminorm of $H^1(\Omega)$. For more information about Sobolev spaces, we refer to the fundamental textbook Adams and Fournier [2003].

Next we state the preliminaries and introduce the calculus notation on surfaces to be used throughout this chapter. The following definitions can be found in Dziuk and Elliott [2013].

Definition 1. *Let $k \in \mathbb{N}$. A subset $\Gamma \subset \mathbb{R}^{n+1}$ is called a $\mathcal{C}^k$ hyper-surface if, for each point $x \in \Gamma$, there exists an open set $\mathcal{U} \subset \mathbb{R}^{n+1}$ containing x and a function $d \in \mathcal{C}^k(\mathcal{U})$ with the property $\nabla d(x) \neq 0, \forall x \in \mathcal{U} \cap \Gamma$, such that Γ can be represented as the zero level-set of the function d*

$$\Gamma := \{x \in \mathcal{U} : d(x) = 0\}. \tag{2.1}$$

Definition 2. *We denote by $T_x\Gamma$ the tangent space to Γ at $x \in \Gamma$ and it is defined as the n-dimensional subspace of $\mathbb{R}^{n+1}$ that is orthogonal to $\nabla d(x)$,*

$$T_x\Gamma = [\nabla d(x)]^{\perp}.$$

Definition 3. *A vector $\boldsymbol{\nu}(x) = (\nu_1, \ldots, \nu_{n+1})^\top \in \mathbb{R}^{n+1}$ is called a unit normal vector at $x \in \Gamma$ if $\boldsymbol{\nu}(x) \perp T_x\Gamma$ and $|\boldsymbol{\nu}(x)| = 1$. Therefore, based on the definition of the tangent space (Definition 2) we have that the outward normal is given by*

$$\boldsymbol{\nu}(x) = \frac{\nabla d(x)}{|\nabla d(x)|}, \tag{2.2}$$

whereas the unit inward normal is given by

$$\boldsymbol{\nu}(x) = -\frac{\nabla d(x)}{|\nabla d(x)|}. \tag{2.3}$$

Definition 4. *Given a function $u : \Gamma \to \mathbb{R}$ which is differentiable in an open neighbourhood of the hyper-surface Γ, the tangential or surface gradient of u, denoted by ∇_Γ, is defined by*

$$\nabla_\Gamma u(x) = \nabla u(x) - (\nabla u(x) \cdot \boldsymbol{\nu}(x))\, \boldsymbol{\nu}(x),$$

where ∇ *denotes the gradient in* $\mathbb{R}^{n+1}$ *and* $x \cdot y$ *denotes the usual scalar product on* $\mathbb{R}^{n+1}$.

The tangential gradient is the projection of the gradient onto the tangent plane. Therefore, $\nabla_\Gamma u \cdot \boldsymbol{\nu} = 0$.

For a vector valued function $\boldsymbol{w} = (w_1, \ldots, w_{n+1})^\top \in \mathbb{R}^{n+1}$ the tangential divergence is defined by

$$\nabla_\Gamma \cdot \boldsymbol{w} = \nabla \cdot \boldsymbol{w} - \sum_{i=1}^{n+1} (\nabla w_i \cdot \boldsymbol{\nu}) \nu_i.$$

Definition 5. *The Laplace – Beltrami operator is defined as the tangential divergence of the tangential gradient*

$$\Delta_\Gamma u(x) = \nabla_\Gamma \cdot \nabla_\Gamma u(x). \tag{2.4}$$

Definition 6. *We define the mean curvature* H *of* Γ *with respect to* $\boldsymbol{\nu}(x)$

$$H(x) = \nabla_\Gamma \cdot \boldsymbol{\nu}(x). \tag{2.5}$$

Let $\mathcal{R}$ an arbitrary portion of Γ. Then, the divergence theorem on a surface Γ is given by

$$\int_{\mathcal{R}} \nabla_{\mathcal{R}} \cdot \boldsymbol{q} \, \mathrm{d}s = -\int_{\mathcal{R}} \boldsymbol{q} \cdot H\boldsymbol{\nu} \, \mathrm{d}s + \int_{\partial\mathcal{R}} \boldsymbol{q} \cdot \boldsymbol{\mu} \, \mathrm{d}s, \tag{2.6}$$

for a vector field $\boldsymbol{q} = (q_1, \ldots, q_{n+1})^\top \in \mathbb{R}^{n+1}$ (Gilbarg and Trudinger [1977]). Here $\boldsymbol{\mu}$ denotes the co-normal vector which is normal to $\partial\mathcal{R}$ and tangent to $\mathcal{R}$. The Green's identity formula on a surface (integration by parts) reads

$$\int_{\mathcal{R}} \nabla_{\mathcal{R}} v \cdot \nabla_{\mathcal{R}} u \, \mathrm{d}s = -\int_{\mathcal{R}} v \Delta_{\mathcal{R}} u \, \mathrm{d}s + \int_{\partial\mathcal{R}} v \nabla_{\mathcal{R}} u \cdot \boldsymbol{\mu} \, \mathrm{d}s, \tag{2.7}$$

for a scalar function v and $\boldsymbol{q} = \nabla_{\mathcal{R}} u$.

Remark 1. *We remark that in (2.6)* $\mathrm{d}s$ *in connection with an integral over* $\mathcal{R}$ *denotes the* n*-dimensional surface measure, while* $\mathrm{d}s$ *in connection with an integral over the* $\partial\mathcal{R}$ *is the* $(n-1)$*-dimensional surface measure.*

2.3 Derivation of reaction-diffusion equations on stationary surfaces

For illustrative purposes, let $\boldsymbol{u}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))^\top$, with $\boldsymbol{x} \in \Gamma$ and $t \in [0, T]$, be a vector of real valued functions corresponding to the concentrations of chemical

species residing on a stationary closed surface Γ. The law of mass conservation can be formulated for an arbitrary portion $\mathcal{R}$ of Γ. From the mass balance conservation law (Murray [2002]), we have for each chemical concentration, that the rate of change of the concentration in a region $\mathcal{R}$ is equal to the sum of the outward flux of the chemical species through the boundary and the net reactive production of the chemicals, within the surface. Hence

$$\frac{d}{dt}\int_{\mathcal{R}} \boldsymbol{u}\,\mathrm{d}s = -\int_{\partial\mathcal{R}} \boldsymbol{q}\cdot\boldsymbol{\mu}\,\mathrm{d}s + \int_{\mathcal{R}} \boldsymbol{f}(\boldsymbol{u})\,\mathrm{d}s, \tag{2.8}$$

where $\boldsymbol{q}$ is the surface flux through the boundary of $\mathcal{R}$, $\boldsymbol{f}(\boldsymbol{u}) = (f_1(\boldsymbol{u}), f_2(\boldsymbol{u}))^\top$ is the net production rate within the surface and $\boldsymbol{\mu}$ the conormal on $\partial\mathcal{R}$. Applying the divergence theorem on the surface Γ (2.6) we replace the boundary surface integral with

$$\int_{\partial\mathcal{R}} \boldsymbol{q}\cdot\boldsymbol{\mu}\,\mathrm{d}s = \int_{\mathcal{R}} \nabla_\Gamma\cdot\boldsymbol{q}\,\mathrm{d}s + \int_{\mathcal{R}} \boldsymbol{q}\cdot\boldsymbol{\nu}H\,\mathrm{d}s. \tag{2.9}$$

We assume that the flux is a tangential vector to $\mathcal{R}$; the components of $\boldsymbol{q}$ normal to $\mathcal{R}$ do not contribute to the flux, i.e. $\boldsymbol{q}\cdot\boldsymbol{\nu} = 0$. Therefore, the last integral vanishes and equation (2.8) becomes

$$\frac{d}{dt}\int_{\mathcal{R}} \boldsymbol{u}\,\mathrm{d}s = -\int_{\mathcal{R}} \nabla_\Gamma\cdot\boldsymbol{q}\,\mathrm{d}s + \int_{\mathcal{R}} \boldsymbol{f}(\boldsymbol{u})\,\mathrm{d}s. \tag{2.10}$$

We take $\boldsymbol{q}$ to be the diffusive flux and thus from the Fick's law (Acheson [1990]), the flux vector $\boldsymbol{q}$ is proportional to the concentration gradient defined as

$$\boldsymbol{q} = -\mathcal{D}\nabla_\Gamma\boldsymbol{u}, \tag{2.11}$$

where $\mathcal{D} = \mathrm{diag}(\mathrm{d}_1, \mathrm{d}_2)$ is the diffusion tensor and $d_i, i = 1, 2$, are positive constant diffusion coefficients. Thus equation (2.10) yields

$$\frac{d}{dt}\int_{\mathcal{R}} \boldsymbol{u}\,\mathrm{d}s = \int_{\mathcal{R}} \nabla_\Gamma\cdot(\mathcal{D}\nabla_\Gamma\boldsymbol{u})\,\mathrm{d}s + \int_{\mathcal{R}} \boldsymbol{f}(\boldsymbol{u})\,\mathrm{d}s. \tag{2.12}$$

Re-arranging equation (2.12) results in

$$\int_{\mathcal{R}} (\boldsymbol{u}_t - \mathcal{D}\Delta_\Gamma\boldsymbol{u} - \boldsymbol{f}(\boldsymbol{u}))\,\mathrm{d}s = 0.$$

Since $\mathcal{R}$ is arbitrary, it follows that the integrand must vanish, i.e.

$$\boldsymbol{u}_t - \mathcal{D}\Delta_\Gamma\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{u}). \tag{2.13}$$

To complete the derivation of the reaction-diffusion equations we need to prescribe boundary and initial conditions. But since the surface Γ is closed no boundary conditions are needed for the system (2.13). Here we impose initial conditions $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$, where $\boldsymbol{u}_0(\boldsymbol{x})$ are positive bounded functions.

One form of non-dimensionalisation of system (2.13) can be written as (Murray [2002]; Madzvamuse et al. [2003]):

$$\hat{\boldsymbol{u}}_t - \hat{\mathcal{D}}\Delta_\Gamma \hat{\boldsymbol{u}} = \gamma \hat{\boldsymbol{f}}(\hat{\boldsymbol{u}}), \tag{2.14}$$

where $\hat{\mathcal{D}} = \mathrm{diag}(1, \mathrm{d})$ and γ a positive scaling parameter. Here $d = \frac{d_2}{d_1}$ represents the ratio of the diffusion coefficients. Furthermore, it can be shown that $\sqrt{\gamma}$ is proportional to the area of the domain and γ measures the relative strength of interaction of the reaction and diffusion. Doubling the value of γ is equivalent to doubling the surface length (Murray [2002]). For notational simplicity we will skip the hats ($\hat{}$) from here on.

Remark 2. *In the case that the surface Γ is open, i.e. $\partial\Gamma \neq \emptyset$, if we consider zero flux boundary conditions (known as homogeneous Neumann boundary conditions) we obtain identical equations. Homogeneous Neumann boundary conditions are suitable for biological pattern formation since these enable self-organisation of the biological process.*

2.4 Reaction kinetics

In this chapter we show the robustness and applicability of the surface finite element method for solving systems of reaction-diffusion equations of the form

$$\boldsymbol{u}_t = \mathcal{D}\Delta_\Gamma \boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}), \tag{2.15}$$

on stationary surfaces. For illustrative purposes we implement the methodology to the well known Schnakenberg or Brusselator model, which is also referred in the literature as *activator-depleted* substrate model (Prigogine and Lefever [1968]; Gierer and Meinhardt [1972]; Schnakenberg [1979]).

This reaction kinetic model is one of the simplest but chemically plausible reactions model for two chemical concentrations $\boldsymbol{u} = (u, v)$, and it is given in non-

dimensional form by

$$\begin{aligned} f_1(u,v) &= \gamma\left(a - u + u^2 v\right), \\ f_2(u,v) &= \gamma\left(b - u^2 v\right), \end{aligned} \tag{2.16}$$

where a and b are nondimensional positive parameters. The biological interpretation of this model is that u and v are produced at constant rates γa and γb respectively. The negative term $-\gamma u$ indicates that u is degraded linearly, the positive nonlinear term represents the activation of u, whereas the negative nonlinear term represents the consumption of the chemical v (Schnakenberg [1979]; Murray [2002]).

Remark 3. *The choice of different types of reaction kinetics for $\boldsymbol{f}(\boldsymbol{u})$ in (2.15) leads to various reaction-diffusion systems with different characterstics. We refer to Maini et al. [1997], Madzvamuse [2000], Murray [2002], for a review of the most commonly used reaction kinetics and their biological meaning.*

2.5 Surface finite element method

We apply the surface finite element method to the reaction-diffusion system with Schnakenberg reaction kinetics

$$\begin{cases} u_t - \Delta_\Gamma u = f_1(u,v) \\ v_t - d\Delta_\Gamma v = f_2(u,v), \end{cases} \tag{2.17}$$

posed on a stationary surface Γ, where $f_1(u,v)$ and $f_2(u,v)$ are given in (2.16).

2.5.1 Weak variational form

Let $\psi \in H^1(\Gamma)$ and $\omega \in H^1(\Gamma)$ be arbitrary test functions. Multiplying the first equation of (2.17) by the test function ψ and integrating over Γ, we obtain

$$\begin{aligned} \int_\Gamma f_1(u,v)\psi \,\mathrm{d}s &= \int_\Gamma u_t\psi \,\mathrm{d}s - \int_\Gamma \psi\Delta_\Gamma u \,\mathrm{d}s \\ &= \int_\Gamma u_t\psi \,\mathrm{d}s - \left(- \int_\Gamma \nabla_\Gamma u \cdot \nabla_\Gamma \psi \,\mathrm{d}s + \int_{\partial\Gamma} \psi \nabla_\Gamma u \cdot \boldsymbol{\mu} \,\mathrm{d}s \right), \end{aligned} \tag{2.18}$$

and similarly multiplying the second equation by ω and integrating we get

$$\begin{aligned} \int_\Gamma f_2(u,v)\omega \,\mathrm{d}s &= \int_\Gamma v_t\omega \,\mathrm{d}s - d\int_\Gamma \omega\Delta_\Gamma v \,\mathrm{d}s \\ &= \int_\Gamma v_t\omega \,\mathrm{d}s - d\left(- \int_\Gamma \nabla_\Gamma v \cdot \nabla_\Gamma \omega \,\mathrm{d}s + \int_{\partial\Gamma} \omega \nabla_\Gamma v \cdot \boldsymbol{\mu} \,\mathrm{d}s \right). \end{aligned} \tag{2.19}$$

Since the surface Γ is closed, $\partial\Gamma = \emptyset$ hence the last term vanishes on each equation.

Remark 4 (Weak variational form)**.** *Given $f_1(u,v)$ and $f_1(u,v)$ by (2.16) and the initial data $u_0 \in H^1(\Gamma)$ and $v_0 \in H^1(\Gamma)$, find $u(\boldsymbol{x},t) \in L^2(0,T;H^1(\Gamma)) \cap H^1(0,T;L^2(\Gamma))$ and $v(\boldsymbol{x},t) \in L^2(0,T;H^1(\Gamma)) \cap H^1(0,T;L^2(\Gamma))$ such that*

$$\begin{cases} \displaystyle\int_\Gamma u_t\psi \,\mathrm{d}s + \int_\Gamma \nabla_\Gamma u \cdot \nabla_\Gamma \psi \,\mathrm{d}s &= \displaystyle\int_\Gamma f_1(u,v)\psi \,\mathrm{d}s, \\ \displaystyle\int_\Gamma v_t\omega \,\mathrm{d}s + d\int_\Gamma \nabla_\Gamma v \cdot \nabla_\Gamma \omega \,\mathrm{d}s &= \displaystyle\int_\Gamma f_2(u,v)\omega \,\mathrm{d}s, \end{cases} \tag{2.20}$$

satisfied for almost every $t \in (0,T)$ and for every $\psi \in H^1(\Gamma)$ and $\omega \in H^1(\Gamma)$.

Remark 5. *Since $H^1(\Gamma) \hookrightarrow L^p(\Gamma)$ for $p < \infty$, $n = 2$ then we have that $u^2 v \in L^2(\Gamma)$ and hence the left hand side of the equation (2.18) and (2.19) is valid (Evans [2009]).*

2.5.2 Space discretisation

We approximate the continuous surface $\Gamma \subset \mathbb{R}^{n+1}$ by a polyhedral approximation Γ_h of Γ, where the nodes $\{X_j\}_{j=1}^N$ of Γ_h lie on Γ. Since the nodes of the polyhedral approximation Γ_h sit on Γ, it can be considered as an interpolation of Γ. Let $\mathcal{T}_h$ be a triangulation of Γ_h consisting of closed simplices S:

$$\Gamma_h := \mathcal{T}_h = \bigcup_{S \in \mathcal{T}_h} S. \tag{2.21}$$

In particular, when $n = 1$ the elements of $\mathcal{T}_h$ (polygonal line) are line segments whereas for $n = 2$ the elements are triangles. When $\Gamma \subseteq \mathbb{R}^3$, the finite triangulated surface $\mathcal{T}_h$ is described by a set of the non-overlapping triangles.

Definition 7. *Taking Γ_h to be a triangulation of a surface Γ embedded in $\mathbb{R}^3$, the mesh size h is defined to be the maximum diameter of the circle inscribed in a triangle S of the polyhedral approximation.*

In order to solve the partial differential equations on a discrete surface Γ_h, we define the surface finite element space

$$S_h := \{\psi_h \in C^0(\Gamma_h) : \psi_h|_S \text{ is linear affine for each } S \in \mathcal{T}_h\}.$$

This is a piecewise linear finite element space on Γ_h and is spanned by the nodal basis functions $\{\chi_1, \chi_2, \ldots, \chi_N\}$ that satisfy the following properties

$$\chi_j \in S_h, \text{ and } \chi_j(X_i) = \delta_{ij},$$

for $j = 1, \ldots, N$.

The semi-discrete weak formulation of (2.20) reads: Find $(u_h, v_h) \in L^2(0, T; S_h)$, such that

$$\begin{cases} \displaystyle\int_{\Gamma_h} u_{h,t}\psi_h \,\mathrm{d}s_h + \int_{\Gamma_h} \nabla_{\Gamma_h} u_h \cdot \nabla_{\Gamma_h} \psi_h \,\mathrm{d}s_h = \int_{\Gamma_h} f_1(u_h, v_h)\psi_h \,\mathrm{d}s_h, \\ \displaystyle\int_{\Gamma_h} v_{h,t}\omega_h \,\mathrm{d}s_h + d\int_{\Gamma_h} \nabla_{\Gamma_h} v_h \cdot \nabla_{\Gamma_h} \omega_h \,\mathrm{d}s_h = \int_{\Gamma_h} f_2(u_h, v_h)\omega_h \,\mathrm{d}s_h, \end{cases} \tag{2.22}$$

for all $\psi_h \in S_h$ and $\omega_h \in S_h$.

Since $(u_h, v_h) \in S_h \times S_h$ and $\{\chi_i\}_{i=1}^N$ are the basis of S_h, we can write u_h as a unique representation of the basis vectors and the nodal values of the vector $\boldsymbol{\alpha_u} = (\alpha_u^1, \ldots, \alpha_u^N)^T$

$$u_h(\boldsymbol{x}, t) = \sum_{i=1}^{N} \alpha_u^i(t)\chi_i(\boldsymbol{x}). \tag{2.23}$$

Similarly,

$$v_h(\boldsymbol{x}, t) = \sum_{i=1}^{N} \alpha_v^i(t)\chi_i(\boldsymbol{x}), \tag{2.24}$$

where $\boldsymbol{\alpha_v} = (\alpha_v^1, \ldots, \alpha_v^N)^T$.

Given (2.23) and (2.24) the system (2.22) transforms into

$$\begin{cases} \displaystyle\int_{\Gamma_h} \left(\sum_{i=1}^{N} \alpha_{u,t}^i \chi_i\right) \psi_h \,\mathrm{d}s_h + \int_{\Gamma_h} \left(\sum_{i=1}^{N} \alpha_u^i \nabla_{\Gamma_h}\chi_i\right) \cdot \nabla_{\Gamma_h}\psi_h \,\mathrm{d}s_h = \int_{\Gamma_h} f_1(u_h, v_h)\psi_h \,\mathrm{d}s_h, \\ \displaystyle\int_{\Gamma_h} \left(\sum_{i=1}^{N} \alpha_{v,t}^i \chi_i\right) \omega_h \,\mathrm{d}s_h + d\int_{\Gamma_h} \left(\sum_{i=1}^{N} \alpha_v^i \nabla_{\Gamma_h}\chi_i\right) \cdot \nabla_{\Gamma_h}\omega_h \,\mathrm{d}s_h = \int_{\Gamma_h} f_2(u_h, v_h)\omega_h \,\mathrm{d}s_h, \end{cases}$$

for all $\psi_h, \omega_h \in S_h$. Taking $\psi_h = \chi_k$ and $\omega_h = \chi_k$, $k = 1, \ldots, N$, in the previous equations, the following system of equations in vector form is obtained

$$\begin{cases} \mathcal{M}\boldsymbol{\alpha}_{u,t} + \mathcal{S}\boldsymbol{\alpha}_u = \boldsymbol{F}_1, \\ \mathcal{M}\boldsymbol{\alpha}_{v,t} + d\mathcal{S}\boldsymbol{\alpha}_v = \boldsymbol{F}_2, \end{cases} \tag{2.25}$$

where $\mathcal{M}$ is the mass matrix whose entries are defined by

$$(\mathcal{M})_{ik} = \int_{\Gamma_h} \chi_j \chi_k \,\mathrm{d}s_h, \tag{2.26}$$

$\mathcal{S}$ is the stiffness matrix with entries

$$(\mathcal{S})_{ik} = \int_{\Gamma_h} \nabla_{\Gamma_h}\chi_j \cdot \nabla_{\Gamma_h}\chi_k \,\mathrm{d}s_h \tag{2.27}$$

and the right hand side describes non-linear vectors whose entries are

$$(\boldsymbol{F}_1)_k = \int_{\Gamma_h} f_1(u_h, v_h)\chi_k \,\mathrm{d}s_h \quad \text{and} \quad (\boldsymbol{F}_2)_k = \int_{\Gamma_h} f_2(u_h, v_h)\chi_k \,\mathrm{d}s_h,$$

respectively.

2.5.3 Time discretisation

We discretise the time interval $[0, T]$ into a finite number of uniform sub-intervals such that

$$0 = t_0 < t_1 < \ldots < t_M = T.$$

Let $\tau > 0$ be the uniform timestep, i.e. $T = M\tau$ for $M \in \mathbb{N}$. We represent by u_h^k and v_h^k the approximate solution at time $t_k = k\tau$, i.e. $u_h^k = u_h(\cdot, t_k)$ and $v_h^k = v_h(\cdot, t_k)$, for $k = 0, \ldots M-1$. The time derivative is approximated by the first order backward Euler scheme

$$u_{h,t} = \frac{u_h^{k+1} - u_h^k}{\tau} \quad \text{and} \quad v_{h,t} = \frac{v_h^{k+1} - v_h^k}{\tau}. \tag{2.28}$$

Time-stepping schemes for reaction-diffusion systems on stationary and evolving domains have been performed by the use of implicit-explicit (IMEX) schemes (Ruuth [1995]; Madzvamuse [2006]). The main feature of these schemes is that the diffusive term is treated implicitly whereas an explicit scheme is used for the approximation of the reaction kinetics. For more details about the different implicit-explicit schemes we refer to Ruuth [1995] and Madzvamuse [2006].

Here we use a modified first-order semi-implicit backward Euler finite difference scheme (1-SBEM), where the diffusion term and the linear reaction terms are treated implicitly while the nonlinear reaction terms are treated semi-implicitly (Madzvamuse [2006]; Madzvamuse [2007]). The nonlinear term $(u_h^{k+1})^2$ is linearised by using a single Picard iteration of the form

$$(u_h^{k+1})^2 \approx u_h^k u_h^{k+1} \tag{2.29}$$

and this is due to the assumption that the two successive approximate solutions at consecutive time steps do not change significantly (Madzvamuse [2006]). Thus, the

fully discrete system of reaction-diffusion equations (2.22) becomes

$$\begin{cases} \left(\frac{1}{\tau}+\gamma\right)\int_{\Gamma_h} u_h^{k+1}\psi_h \,\mathrm{d}s_h + \int_{\Gamma_h} \nabla_{\Gamma_h} u_h^{k+1}\cdot\nabla_{\Gamma_h}\psi_h \,\mathrm{d}s_h \\ \qquad\qquad -\gamma\int_{\Gamma_h} u_h^k v_h^k u_h^{k+1}\psi_h \,\mathrm{d}s_h = \frac{1}{\tau}\int_{\Gamma_h} u_h^n\psi_h \,\mathrm{d}s_h + \gamma a\int_{\Gamma_h}\psi_h \,\mathrm{d}s_h, \\ \\ \frac{1}{\tau}\int_{\Gamma_h} v_h^{k+1}\omega_h \,\mathrm{d}s_h + d\int_{\Gamma_h} \nabla_{\Gamma_h} v_h^{k+1}\cdot\nabla_{\Gamma_h}\omega_h \,\mathrm{d}s_h \\ \qquad\qquad +\gamma\int_{\Gamma_h} (u_h^{k+1})^2 v_h^{k+1}\omega_h \,\mathrm{d}s_h = \frac{1}{\tau}\int_{\Gamma_h} v_h^k\omega_h \,\mathrm{d}s_h + \gamma b\int_{\Gamma_h}\omega_h \,\mathrm{d}s_h, \end{cases}$$

for all $(\psi_h, \omega_h) \in S_h \times S_h$. Using matrix representation we have

$$\begin{cases} \left(\left(\frac{1}{\tau}+\gamma\right)\mathcal{M} + \mathcal{S} - \gamma\mathcal{K}_1\right)\boldsymbol{\alpha}_u^{k+1} = \frac{1}{\tau}\mathcal{M}\boldsymbol{\alpha}_u^k + \boldsymbol{F}_1, \\ \left(\frac{1}{\tau}\mathcal{M} + d\mathcal{S} + \gamma\mathcal{K}_2\right)\boldsymbol{\alpha}_v^{k+1} = \frac{1}{\tau}\mathcal{M}\boldsymbol{\alpha}_v^k + \boldsymbol{F}_2, \end{cases} \tag{2.30}$$

for $k = 0, \ldots, M$ where $\mathcal{M}$ is the mass matrix, $\mathcal{S}$ the stiffness matrix as defined before in (2.26) and (2.27), and $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ are forcing vectors whose entries are given by

$$(\boldsymbol{F}_1)_i = \gamma a\int_{\Gamma_h}\chi_i \,\mathrm{d}s_h, \quad \text{and} \quad (\boldsymbol{F}_2)_i = \gamma b\int_{\Gamma_h}\chi_i \,\mathrm{d}s_h.$$

The matrices $\mathcal{K}_1$ and $\mathcal{K}_2$ have elements given by

$$(\mathcal{K}_1)_{ij} = \int_{\Gamma_h} u_h^k v_h^k \chi_i\chi_j \,\mathrm{d}s_h, \quad \text{and} \quad (\mathcal{K}_2)_{ij} = \int_{\Gamma_h} (u_h^{k+1})^2\chi_i\chi_j \,\mathrm{d}s_h,$$

and these are associated with the nonlinear reactions. The formulation (2.30) results in a linear algebraic problem posed in matrix form as

$$\begin{pmatrix} \left(\frac{1}{\tau}+\gamma\right)\mathcal{M} + \mathcal{S} - \gamma\mathcal{K}_1 & \mathbf{0} \\ \mathbf{0} & \frac{1}{\tau}\mathcal{M} + d\mathcal{S} - \gamma\mathcal{K}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha_u}^{k+1} \\ \boldsymbol{\alpha_v}^{k+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\tau}\mathcal{M}\boldsymbol{\alpha_u}^k + \boldsymbol{F}_1 \\ \frac{1}{\tau}\mathcal{M}\boldsymbol{\alpha_v}^k + \boldsymbol{F}_2 \end{pmatrix},$$

for $k = 0, \ldots, M$.

Remark 6. *Alternatively, (2.22) could be solved using a Newton's method. Madzvamuse and Chung [2014] have shown that the Newton method is an alternative method for treating the nonlinear reaction kinetics. In particular, a single Newton iteration was shown to be sufficient to the accuracy of the results and could become crucially important as an aid for large computational savings.*

2.6 Numerical results

This section is devoted to approximating solutions of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics on stationary surfaces, in order to illustrate the effectiveness, robustness and applicability of the surface finite element method.

We choose and keep constant a set of parameter values for a, b and d, under which patterns are formed. Varying the parameter γ, we examine how it affects pattern formation. In all our simulations we fix the parameters $a = 0.1$, $b = 0.9$ and the diffusion coefficient $d = 10$. With these parameter values, it can easily be shown that system (2.30) admits a homogeneous steady state $(1, 0.9)$. We prescribe initial conditions as small random perturbations around the uniform homogeneous steady state of the order $\sim 10^{-2}$. In all the simulations the following stopping criteria were enforced:

$$\frac{||u^{k+1} - u^k||}{\tau} \leq 10^{-4} \quad \text{and} \quad \frac{||v^{k+1} - v^k||}{\tau} \leq 10^{-4}. \tag{2.31}$$

2.6.1 Experimental order of convergence

In this section we show that the approximate solution obtained by the method described in this chapter converges to an "exact" solution as we refine the mesh size. Since there is no analytical solution to the reaction-diffusion (2.17) system with Schnakeberg reaction kinetics, we construct a solution that will satisfy a modified version of (2.17). In the following example, we consider the surface Γ to be the unit sphere. Let define

$$\zeta(\boldsymbol{x}, t) = e^{-2t} x_1 \text{ and } \xi(\boldsymbol{x}, t) = e^{-2t} x_2$$

and $u(\boldsymbol{x}, t) = \zeta(\boldsymbol{x}, t)$ and $v(\boldsymbol{x}, t) = \xi(\boldsymbol{x}, t)$ be the exact solution of the following system

$$\begin{cases} u_t - \Delta_\Gamma u - \gamma(-u + u^2 v) = \zeta_t - \Delta_\Gamma \zeta - \gamma(a - \zeta + \zeta^2 \xi) \\ v_t - d\Delta_\Gamma v + \gamma u^2 v = \xi_t - d\Delta_\Gamma \xi + \gamma(b - \zeta^2 \xi). \end{cases} \tag{2.32}$$

Since the surface Γ is closed no boundary conditions are needed and the initial conditions are defined by

$$u_0(\boldsymbol{x}, 0) = x_1 \text{ and } v_0(\boldsymbol{x}, 0) = x_2.$$

For this experiment we take $a = 0.1$, $b = 0.9$, $d = 10$ and $\gamma = 60$. Also, we choose final time $T_f = 10$ and a time step $\tau \approx h^2$, where h is the mesh size. Solving the (2.32) using the finite element scheme on the different triangulations we are able to calculate the error from the exact solution $\zeta(\boldsymbol{x}, t)$ at each time step. Then, the error between the exact solution $\zeta(\boldsymbol{x}, t)$ and the approximate solution (u_h, v_h) is given by

$$e_u = ||u(\boldsymbol{x}, t) - \zeta(\boldsymbol{x}, t)||_{L^2(\Gamma)} \quad \text{and} \quad e_v = ||v(\boldsymbol{x}, t) - \zeta(\boldsymbol{x}, t)||_{L^2(\Gamma)}. \tag{2.33}$$

Let denote by e_u^i and e_u^{i-1} the two different errors at two different mesh sizes h_i and h_{i-1}, we calculate the experimental order of convergence (eoc) for the u-variable by

$$(eoc)_u^i = \frac{\log(e_u^i) - \log(e_u^{i-1})}{\log(h_i) - \log(h_{i-1})}, \tag{2.34}$$

for $i = 1, \ldots, 8$. Similarly we can define the experimental order of convergence for the v-variable. In Table 2.1 we report on the convergence results.

Experiment	h (mesh size)	e_u	eoc_u	e_v	eoc_v
1	1.414214	0.2354	–	0.2934	–
2	1.001031	0.0563	4.129325	0.0740	3.972321
3	0.832729	0.0225	5.010216	0.0306	4.832545
4	0.434275	0.0067	1.869174	0.0094	1.812932
5	0.219448	0.0018	1.932472	0.0026	1.906149
6	0.110015	$4.5328{\cdot}10^{-4}$	1.981453	$6.4731{\cdot}10^{-4}$	1.989376
7	0.055044	$1.1360{\cdot}10^{-4}$	1.996237	$1.6170{\cdot}10^{-4}$	2.001031
8	0.036163	$4.9200{\cdot}10^{-5}$	1.995358	$6.9963{\cdot}10^{-5}$	1.997632

Table 2.1: Experimental order of convergence (eoc) for the example from §2.6.1

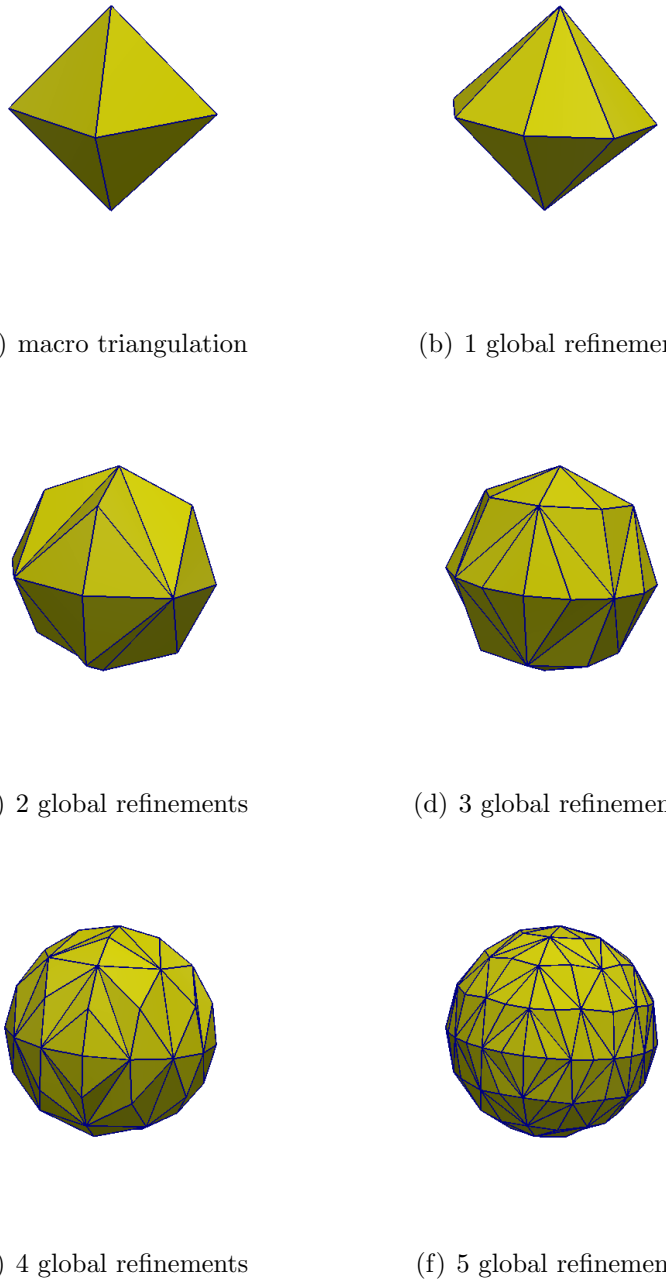
2.6.2 Computer simulations on a stationary sphere

For this section, the surface Γ is taken to be a sphere of radius one. The method requires a triangulation of the initial surface. An initial coarse, or macro, triangulation Γ_h is constructed by hand using a few grid points on the sphere. This initial approximation (macro triangulation) of the surface Γ has 8 large elements with 6 vertices sitting on the unit sphere (see Figure 2.2 (a)). This triangulation is then refined uniformly using the bisection method and the new vertices are projected onto

the surface of the surface Γ. This process is implemented in ALBERTA (Schmidt and Siebert [2005]).

In all the following simulations, we present a triangulation Γ_h of the unit sphere with 16386 vertices and 32768 elements and select a fixed time step $\tau = 1 \times 10^{-3}$. Figure 2.3 shows the solutions of u and v corresponding to the reaction-diffusion (2.17) system with Schnakenberg reaction kinetics when $\gamma = 200$. Figure 2.4 depicts the convergence history for the u and v variable, respectively. We note that regions with high concentrations of the chemical u correspond to regions of low concentrations of the chemical v and vice versa. Also, it can be detected that since the sphere has constant curvature throughout the surface, the patterns are distributed uniformly.

Remark 7. *Figure 2.4 shows that the numerical method is converging but does not show the order of convergence. However, we have not dedicated to study error analysis of such models in this thesis.*



(a) macro triangulation

(b) 1 global refinement

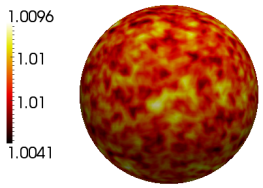
(c) 2 global refinements

(d) 3 global refinements

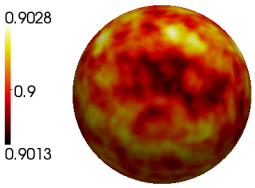
(e) 4 global refinements

(f) 5 global refinements

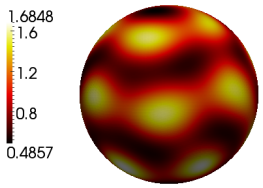
Figure 2.2: Successive global refinements. The initial triangulation is refined uniformly using the bisection method. This process is implemented in ALBERTA (Schmidt and Siebert [2005])



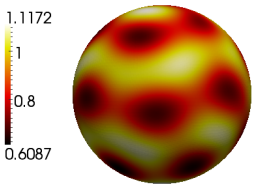
(a) $t = 0$



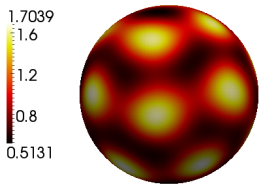
(b) $t = 0$



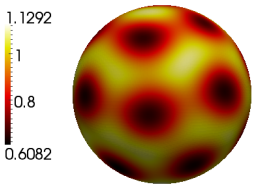
(c) $t = 1$



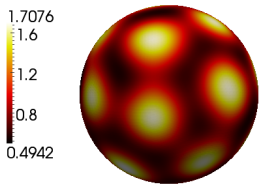
(d) $t = 1$



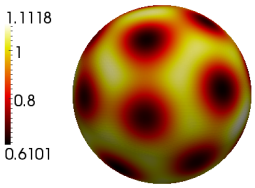
(e) $t = 4$



(f) $t = 4$

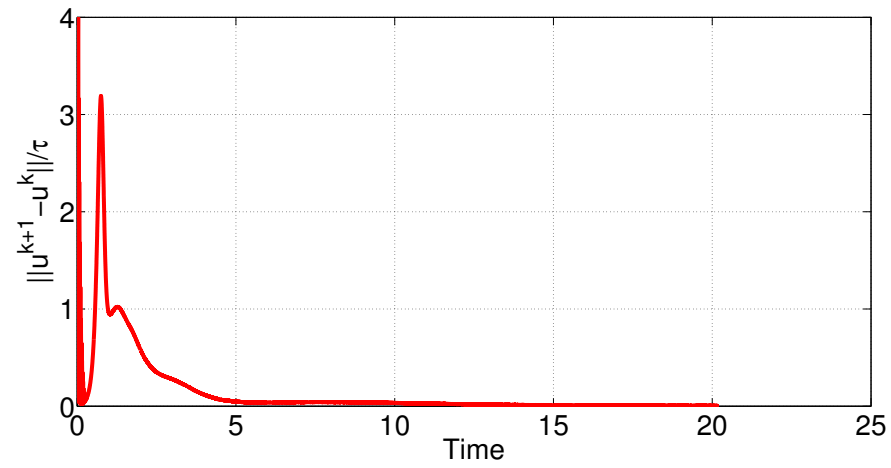


(g) $t \approx 20$

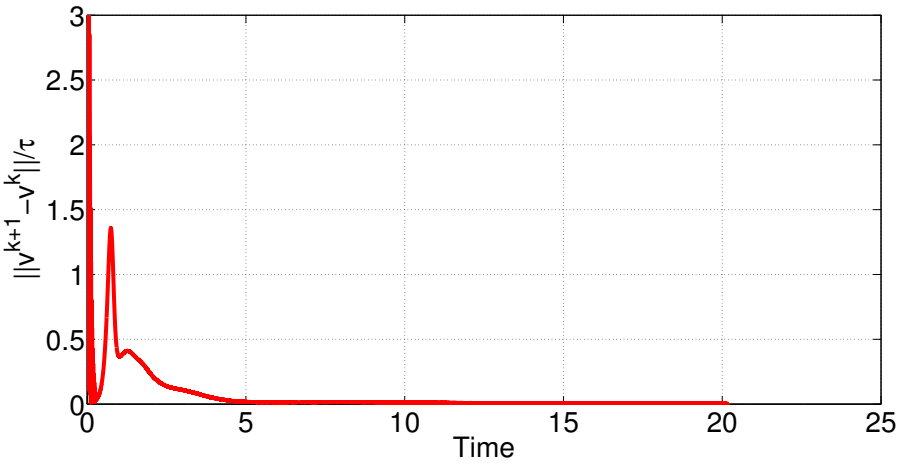


(h) $t \approx 20$

Figure 2.3: Numerical solutions corresponding to the chemical species u (left column) and v (right column) of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics, for $\gamma = 200$. For parameter and numerical values, see Table 2.2 (Experiment 3). Spots and stripes patterns emerge initially which evolve finally into stable spot patterns.



(a)



(b)

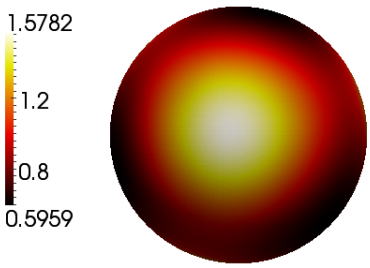
Figure 2.4: The history evolution of the L_2 error norm between successive numerical solutions corresponding to the chemical species (a) u and (b) v of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics for $\gamma = 200$. For parameter and numerical values, see Table 2.2 (Experiment 3).

The effects of γ on pattern generation

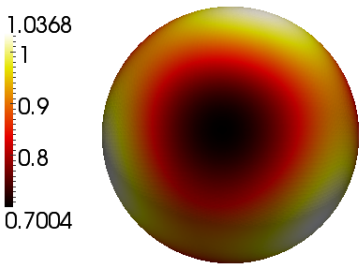
In this section, we investigate the effect of varying the parameter γ on the emergence of patterns. Figures 2.5 and 2.6 display the solutions for the concentrations u and v of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics. The parameters values for the five experiments are given in Table 2.2. We can see in Figures 2.5 and 2.6, that as the scale factor γ increases, the number of spots increases. Similar results are reported in Barreira et al. [2011].

Experiment	$\boldsymbol{\gamma}$	Nr. of elements	timestep
1	**29**	32768	1×10^{-3}
2	**60**	32768	1×10^{-3}
3	**200**	32768	1×10^{-3}
4	**500**	32768	1×10^{-3}
5	**800**	32768	1×10^{-3}

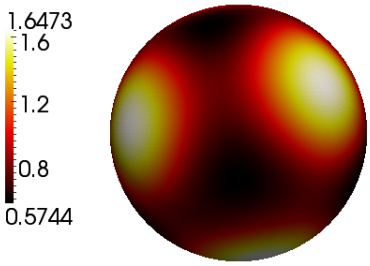
Table 2.2: Parameter values used for the solution of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics on the unit sphere.



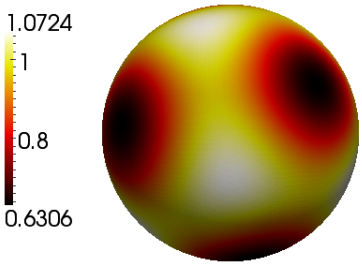
(a) $\gamma = 29$



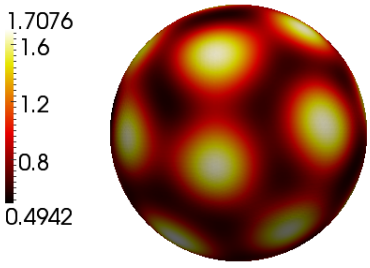
(b) $\gamma = 29$



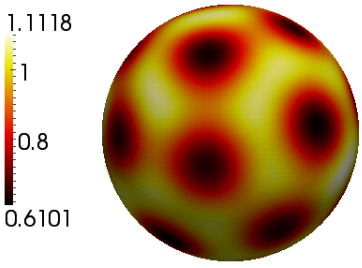
(c) $\gamma = 60$



(d) $\gamma = 60$

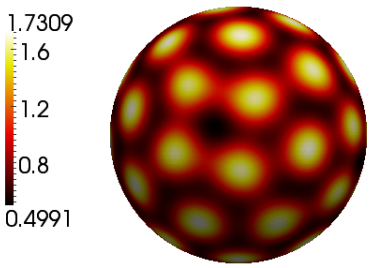


(e) $\gamma = 200$

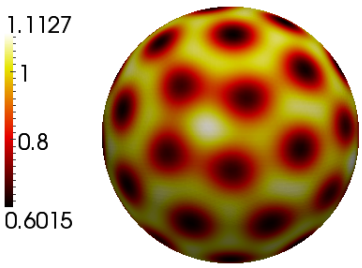


(f) $\gamma = 200$

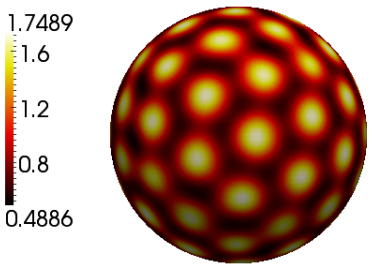
Figure 2.5: Numerical solutions corresponding to the chemical species u (left column) and v (right column) of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics, for different values of γ. For parameter and numerical values, see Table 2.2. As γ increases the number of spots increases.



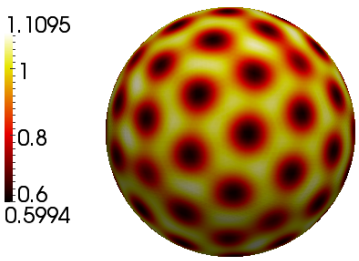
(a) $\gamma = 500$



(b) $\gamma = 500$



(c) $\gamma = 800$



(d) $\gamma = 800$

Figure 2.6: Numerical solutions corresponding to the chemical species u (left column) and v (right column) of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics, for different values of γ. For parameter and numerical values, see Table 2.2. As γ increases the number of spots increases.

2.6.3 Pattern formation on stationary surfaces of neutrophils

The main focus of this section is to illustrate the generality of the surface finite element method on realistic surfaces, where the surfaces are **irregular** and complex. In addition to this we examine the role of the surface geometry on pattern formation. As the aim of this thesis was motivated by the study and understanding neutrophil cell migration *in vivo*, we solve surface partial differential equations on their surfaces (although they are not evolving). Domain evolution will be considered in future studies.

Mesh generation

The macro triangulation Γ_h of each surface of neutrophil (see Figure 2.7) is obtained by using the marching cube algorithm (Lorensen and Cline [1987]; Marchet [2012]). The marching cube algorithm is a technique for constructing triangulated surfaces from 3D medical data. For details about the technique we refer to Lorensen and Cline [1987] and Marchet [2012].

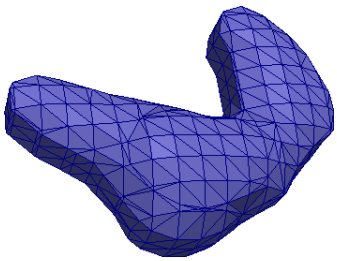


Figure 2.7: Three – dimensional macro triangulation (Γ_h) of the stationary surface of a neutrophil, which is obtained using the marching cube algorithm (Lorensen and Cline [1987]).

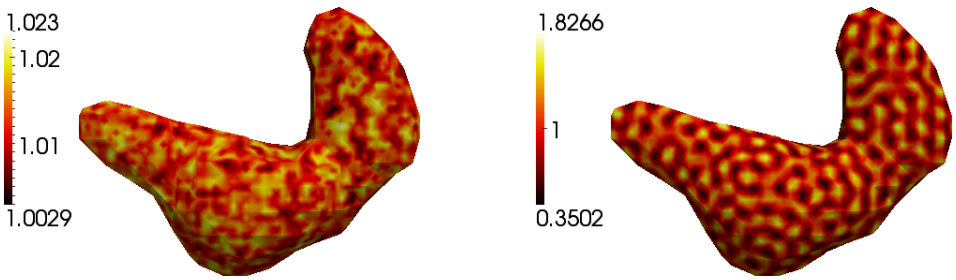
Remark 8 (Triangulated surfaces of the neutrophils)**.** *All the triangulated surfaces that have been used in the following numerical experiments have been generated by Marchet [2012].*

Computer simulations

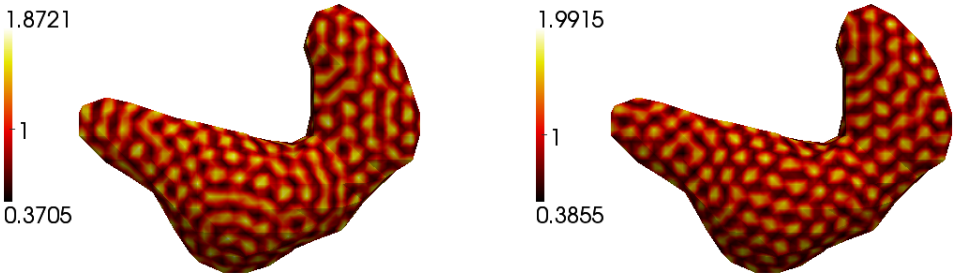
The results of the simulations on different surfaces are shown in Figures 2.8-2.10. In the presentation of the numerical simulations, the chemical concentrations of v have been omitted as they are 180° out of phase with those of u. From the results, we see a combination of stripe and spot patterns. The surface geometry seems to play a pivotal role in the formation of patterns. Here, in contrast to the results on the stationary sphere, the patterns are not distributed uniformly and this may be due to the surface geometry. It is clear that curvature influences the formation of patterns with different wavelengths. We observe that the complexity of a surface geometry (such as the surfaces of the neutrophils) affects the pattern formation.

Experiment	γ	Nr. of elements	timestep	Figure
1	**200**	12352	5×10^{-3}	2.8
2	**200**	12544	5×10^{-3}	2.9
3	**200**	12992	5×10^{-3}	2.10

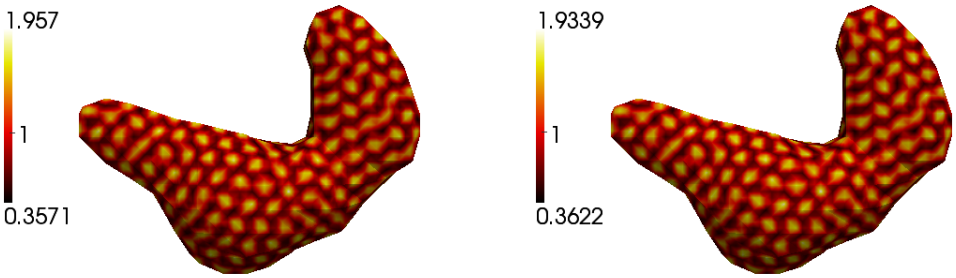
Table 2.3: Parameter values used for the solution of the reaction-diffusion system (2.17) with Schnakenberg reaction kinetics on different stationary surfaces of neutrophils.



(a) $t = 0$ (b) $t = 1$

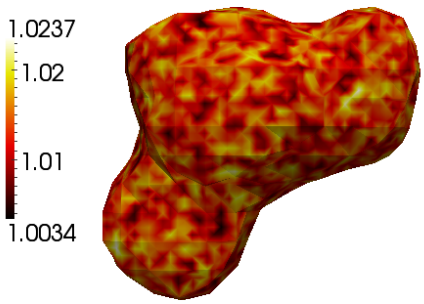


(c) $t = 2$ (d) $t = 10$

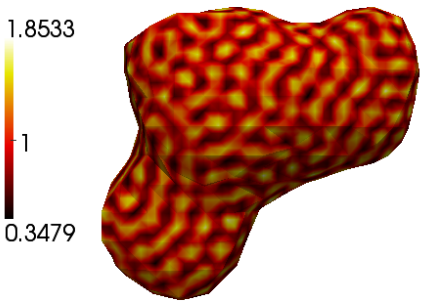


(e) $t = 88$ (f) $t = 176$

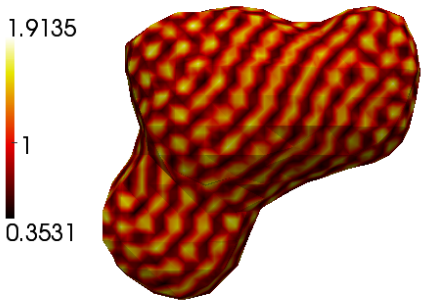
Figure 2.8: Numerical solutions corresponding to the u-chemical species of the reaction-diffusion system (2.17) on a stationary surface of a neutrophil with 12352 elements. Numerical parameter values are given in Table 2.3 (Experiment 1).



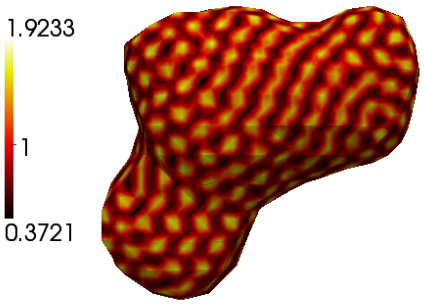
(a) $t = 0$



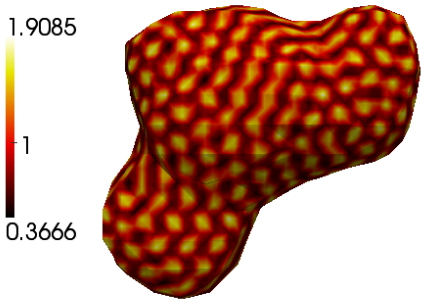
(b) $t = 1$



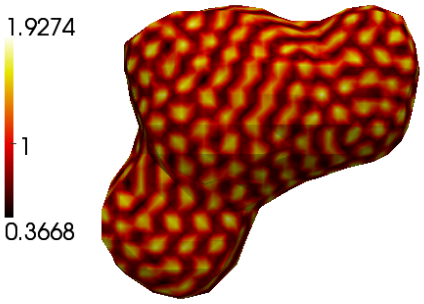
(c) $t = 10$



(d) $t = 20$



(e) $t = 60$



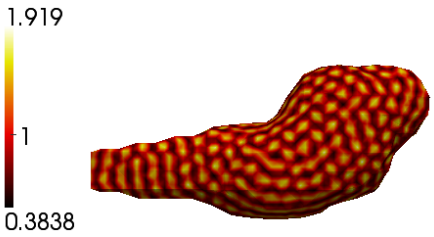
(f) $t = 154$

Figure 2.9: Numerical solutions corresponding to the u-chemical species of the reaction-diffusion system (2.17) on a stationary surface of a neutrophil with 12544 elements. Stripe and spot patterns are observed. Numerical parameter values are given in Table 2.3 (Experiment 2).

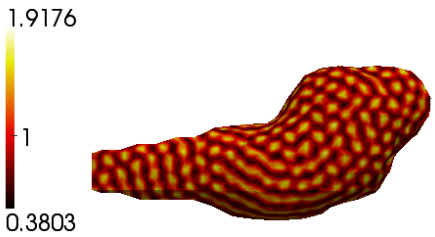
(a) $t = 0$

(b) $t = 1$

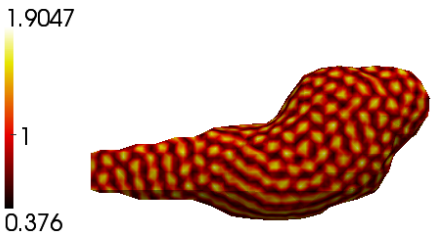
(c) $t = 10$



(d) $t = 30$

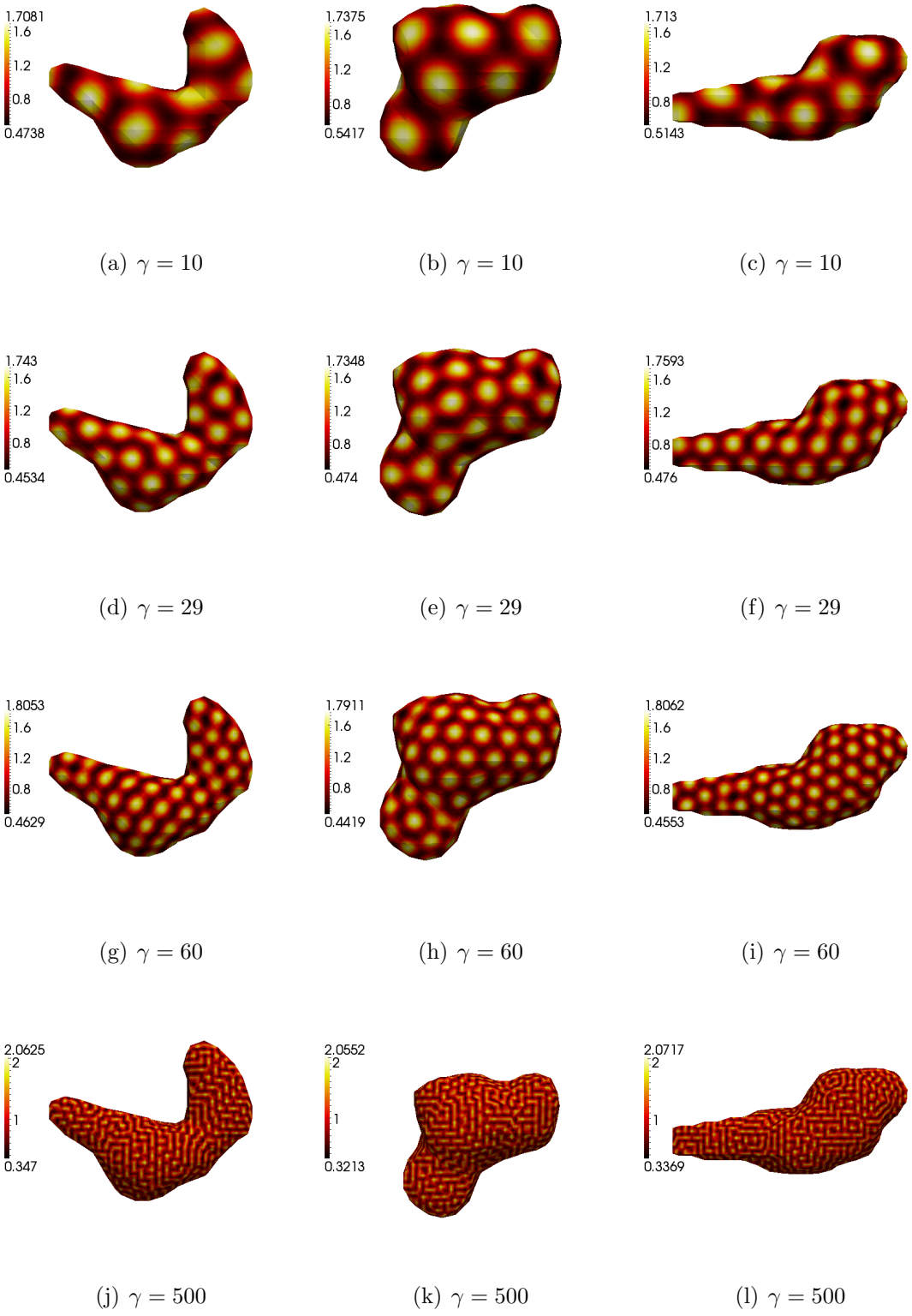


(e) $t = 72$



(f) $t = 144$

Figure 2.10: Numerical solutions corresponding to the u-chemical species of the reaction-diffusion system (2.17) on a stationary surface of a neutrophil with 12992 elements. Stripe, spot and circular patterns are observed. Numerical parameter values are given in Table 2.3 (Experiment 3).



(a) $\gamma = 10$ (b) $\gamma = 10$ (c) $\gamma = 10$

(d) $\gamma = 29$ (e) $\gamma = 29$ (f) $\gamma = 29$

(g) $\gamma = 60$ (h) $\gamma = 60$ (i) $\gamma = 60$

(j) $\gamma = 500$ (k) $\gamma = 500$ (l) $\gamma = 500$

Figure 2.11: Numerical solutions, at the final time of the simulations, corresponding to the u-chemical species of the reaction-diffusion system (2.17) on different stationary surfaces of neutrophils when the parameter γ is varied.

2.7 Conclusion

Modelling and simulation of reaction-diffusion systems on arbitrary surfaces is emerging as a very popular research field with a growing number of applications in developmental biology, cancer research, bio-medical engineering and cell motility (Neilson et al. [2010]; Neilson et al. [2011a]; Barreira et al. [2011]; Elliott et al. [2012]). In this chapter we introduced the surface finite element method that allows us to solve efficiently and robustly reaction-diffusion systems on stationary surfaces. For illustrative purposes we used the well-known reaction-diffusion system with Schnakenberg reaction kinetics in all of our simulations. Our results strongly support the following observations:

- By taking a fixed set of parameter values and changing only the surface geometry, we observe the emergence of wide range of patterns; either spots, stripes or a combination of these. The results of this chapter lead to the observation that the surface geometry and the curvature play a vital role in pattern formation. Recently, Venkataraman et al. [2011] investigated the formation of patterns on the skin of the Amago trout, from its early larval stages to its adulthood and have shown that the curvature of a surface could influence the pattern mechanism during growth development. Similar observations regarding the role of the curvature are highlighted in Varea et al. [1999], Plaza et al. [2004], Landsberg and Voigt [2010].

- The results from our numerical simulations confirm that the parameter γ influences the emergence of pattern complexity.

Demonstrating the performance and applicability of the surface finite element method provides us with a foundation to tackle the following problems in future studies:

- The rapid development in cell microscopy has generated a huge amount of discrete imaging data of migrating cells. This progress can be used to develop a mathematical and computational framework for describing experimental observations using the theory of pattern formation on appropriate biological surfaces. For example, by proposing mathematical models for cell motility, could

model the spatiotemporal dynamics associated with cell migration by coupling systems of reaction-diffusion equations on surfaces of evolving cells. This framework necessitates modelling evolution laws for the cell motion. This law will incorporate cell/surface features such as shape and curvature, and will take into account the bio-physics underlying this process. The solutions of these phenomenological mathematical models will be a step towards the understanding and the analysis of the mechanisms that govern cell motility.

In the following chapters, we describe a mathematical framework based on an optimal control approach for cell tracking with no information regarding the biochemistry of the cells. The proposed method lies in the fact that the model which drives the tracking procedure is a simplification of recently derived physically motivated models for cell motility. This approach can be generalised and extended by using mathematical models as described above which include more biological aspects. Thus, a robust and efficient candidate numerical method for the solution of such problems would be the surface finite element method.

- So far the mathematical models that are used in modelling of cell movement are posed either on the surface (cell membrane) or in the interior of the cell. The bulk-surface finite element method is introduced recently, by Elliott and Ranner [2012] with potential applications especially in cell motility. Thus, a more challenging task for the study of cell motility is to derive new models that will couple the internal cell dynamics to cell surface dynamics. Therefore this framework is a stepping stone in this direction.

Chapter 3

Interpolation method

3.1 Introduction

In this chapter, a general framework based on cubic spline interpolation is presented to introduce a simple single cell tracking algorithm. In Madzvamuse et al. [2003] cubic spline interpolation was applied between successive given positions of the boundary nodes of a deforming shape. Here in a similar way, by using cubic splines, a series of intermediate cell-surface boundaries are generated that describe and track the cell surface evolution, from static image data-sets.

In many practical fields such as medicine, physics and engineering the amount of data obtained through experimental or statistical studies is usually very large and the use of interpolation methods sometimes becomes necessary to correlate and analyse these data. In addition, interpolation methods are commonly used in financial markets (Hagan and West [2006]).

Cubic spline interpolation is broadly used to achieve a smooth continuous function that passes through a given discrete data set. Cubic splines among other interpolation schemes have attracted the attention of many researchers and they have become a useful tool in such fields as computer graphics and image processing, where smoothness is of vital importance in many applications (Miklos [2004]). Furthermore, cubic splines prove to be efficient in medical image processing, for instance, where the medical image magnification or rotation plays a pivotal role in diagnosis and treatment (Lehmann et al. [1999]).

The structure of this chapter is as follows: We start by describing in Section 3.2

the cubic spline interpolation method and deriving the appropriate equations. Then, in Section 3.3 we introduce our approach to a simple single cell tracking algorithm based on cubic interpolation and illustrate some of our numerical results applied to two-dimensional curves of neutrophils as observed in zebrafish *Danio rerio larvae*. Finally, in Section 3.4 we present some conclusions of our approach.

3.2 Interpolation methods

Interpolation is a process of estimating a new set of data points from a given set of discrete data points. The new data points must be within the range of the original set.

There is a multitude of different interpolation methods (De Boor [1978]; Iwashita [2013]; Kreyszig [2007]). Linear interpolation is one of the simplest method of interpolating given discrete data. This method results in a straight line connecting two successive original data points and consists of a first order polynomial.

In this chapter, we will instead focus on the cubic spline interpolation. The main idea of the cubic spline interpolation is to construct smooth curves through a number of given data points. The interpolant is constructed such that it is smooth in the sense of having continuous first and second derivatives at each data point.

3.2.1 Cubic spline interpolation

We now describe the procedure of constructing a cubic spline interpolant $P(t) = (x = x(t), y = y(t))$ which interpolates a series of points $(x_i = x(t_i), y_i = y(t_i))$ for $i = 1, \ldots, n$, based on Press et al. [1990]. $P(t)$ is a piecewise polynomial function that consists of $n - 1$ cubic polynomials $P_k(t)$ defined on each interval $[t_k, t_{k+1}]$, $k = 1, \ldots, n - 1$. The cubic spline interpolant, $P_k(t)$ is constructed based on the following conditions

(a) Each piecewise spline passes through all the data points, such that:

$$P_k(t_{k+1}) = P_{k+1}(t_{k+1}), \tag{3.1}$$

where $P_{k+1}(t_{k+1})$ indicates the cubic function defined on the interval $[t_{k+1}, t_{k+2}]$.

(b) The first derivative with respect to time t should be continuous across the boundary between any two intervals, such that

$$P'_k(t_{k+1}) = P'_{k+1}(t_{k+1}), \tag{3.2}$$

where $P'(t)$ is defined by $P'(t) = (x'(t), y'(t))$.

(c) The second derivative with respect to time t is continuous for both cubic functions on either side of a point

$$P''_k(t_{k+1}) = P''_{k+1}(t_{k+1}), \tag{3.3}$$

where $P''(t)$ is defined by $P''(t) = (x''(t), y''(t))$.

For illustrative purposes, let us consider the derivation of $P_j(t)$ at the time interval $[t_j, t_{j+1}]$ for any $j = 1, \ldots, n-1$. After applying a linear interpolation between the data point on the interval $[t_j, t_{j+1}]$ we have

$$(x(t), y(t)) = A(t)(x_j, y_j) + B(t)(x_{j+1}, y_{j+1}), \tag{3.4}$$

where

$$A(t) = \frac{t_{j+1} - t}{t_{j+1} - t_j} \quad \text{and} \quad B(t) = 1 - A(t) = \frac{t - t_j}{t_{j+1} - t_j}. \tag{3.5}$$

The second derivative with respect to time t of equation (3.4) is zero within the time interval and it is undefined at the boundaries. This comes in contrast with the assumption of the cubic spline interpolation, where the first derivative is smooth and the second derivative is continuous, both within an interval and its boundaries.

Now, suppose that in addition to the data points $(x(t_k), y(t_k))$, the values of the second derivatives $(x''(t_k), y''(t_k))$ for $k = 1, \ldots, n$, are given. In order to have the desired continuity to the second derivative we can add to the right hand side of (3.4) a cubic polynomial whose second derivative fluctuates linearly from $x''(t_j)$ to $x''(t_{j+1})$, within each interval $[t_j, t_{j+1}]$. Also, we construct this cubic polynomial to have zero values at each node t_j and t_{j+1}. Then, it can be shown that the interpolant on the interval $[t_j, t_{j+1}]$, can be rewritten as a cubic polynomial with respect time t as

$$(x(t), y(t)) = A(t)(x_j, y_j) + B(t)(x_{j+1}, y_{j+1}) + C(t)(x''_j, y''_j) + D(t)(x''_{j+1}, y''_{j+1}), \tag{3.6}$$

where the coefficients $C(t)$ and $D(t)$ are given by

$$C(t) = \frac{1}{6}(A(t)^3 - A(t))(t_{j+1} - t_j)^2 \quad \text{and} \quad D(t) = \frac{1}{6}(B(t)^3 - B(t))(t_{j+1} - t_j)^2 \tag{3.7}$$

for $j = 1, \ldots, n-1$.

Using the assumption of continuity of the first derivative across the boundary between the two intervals $[t_{j-1}, t_j]$ and $[t_j, t_{j+1}]$, results in the following system of equations

$$\begin{aligned} \frac{t_j - t_{j-1}}{6}(x''_{j-1}, y''_{j-1}) + \frac{t_{j+1} - t_{j-1}}{3}(x''_j, y''_j) + \frac{t_{j+1} - t_j}{6}(x''_{j+1}, y''_{j+1}) \\ = \frac{(x_{j+1}, y_{j+1}) - (x_j, y_j)}{t_{j+1} - t_j} - \frac{(x_j, y_j) - (x_{j-1}, y_{j-1})}{t_j - t_{j-1}}, \end{aligned} \tag{3.8}$$

for $j = 2, \ldots, n-1$. This is a tridiagonal system of $n-2$ equations for n unknowns $\Big((x''(t_1), y''(t_1)), (x''(t_2), y''(t_2)), \ldots, (x''(t_n), y''(t_n))\Big)$.

For the calculation of a unique cubic spline, two additional conditions must be imposed at each end, t_1 and t_n. A possible choice is the co-called natural spline end conditions:

$$\begin{cases} (x''(t_1), y''(t_1)) &= (0,0), \\ (x''(t_n), y''(t_n)) &= (0,0). \end{cases} \tag{3.9}$$

We take equation (3.6) and combine with boundary conditions (3.9) to give a linear system of n equations for the n unknowns $(x''(t_k), y''(t_k))$ for $k = 1, \ldots, n$. Solving this tridiagonal system of linear equations, condition (c) is satisfied and substituting back into the equation (3.6) the cubic spline interpolant is obtained.

3.3 Numerical results

In this section we apply the cubic spline interpolation on a given set of experimental observations from the neutrophils of zebrafish *Danio rerio larvae* to generate a series of intermediate cell positions. The cubic spline interpolation is constructed such that the point trajectories are smooth (C^2). The experimental observations represent the cell membrane of the neutrophil and each cell membrane of the neutrophil is given by a discrete sequence of its boundary points. In particular, $\Gamma(t_k) := (\boldsymbol{x}(t_k), \boldsymbol{y}(t_k))$ where $\boldsymbol{x}(t_k) \in \mathbb{R}^M$ and $\boldsymbol{y}(t_k) \in \mathbb{R}^M$ consist of the x and y coordinate of the vertices from the given set of the boundary points of each cell membrane, for $k = 0, \ldots, 7$

and in the following experiments we are taking $M = 180$. Thus, each cell membrane $(\Gamma(t_k))$ of the neutrophil is a polygonal line, which vertices are the points $(x(t_k)_i, y(t_k)_i) \in \mathbb{R}^2$, for $k = 0, \ldots, 7$ and $i = 1, \ldots, M$, and its (undirected) edges are defined by subsequent vertices, additionally, connecting the last with the first vertex (see Figure 3.1). In order to apply the cubic spline interpolation we need

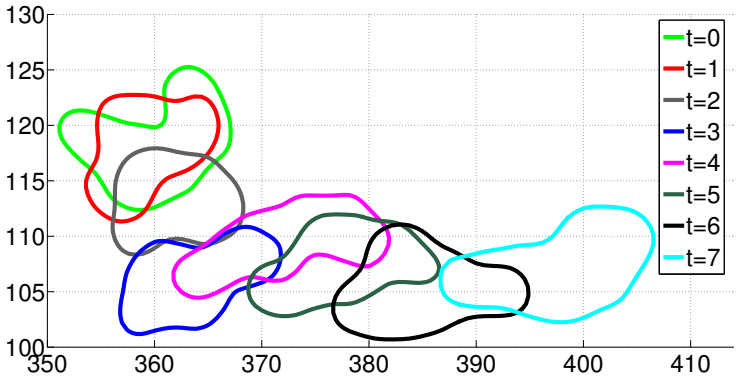


Figure 3.1: Experimental observations representing a discrete set of the locations of the cell membrane corresponding to the *in vivo* migration of a single neutrophil as observed in the zebrafish neutrophils from time $t = 0$ to time $t = 7$ (Henry et al. [2013b]).

to define a mapping of every point through two given successive cell membrane observations. The point, which will define uniquely the entire mapping between two successive polygonal lines $\Gamma(t_k)$ and $\Gamma(t_{k+1})$, $k = 0, \ldots, 6$, is chosen to be the one with the maximum of the minimum distances between each point of the first and second polygonal line, respectively. This can be described by

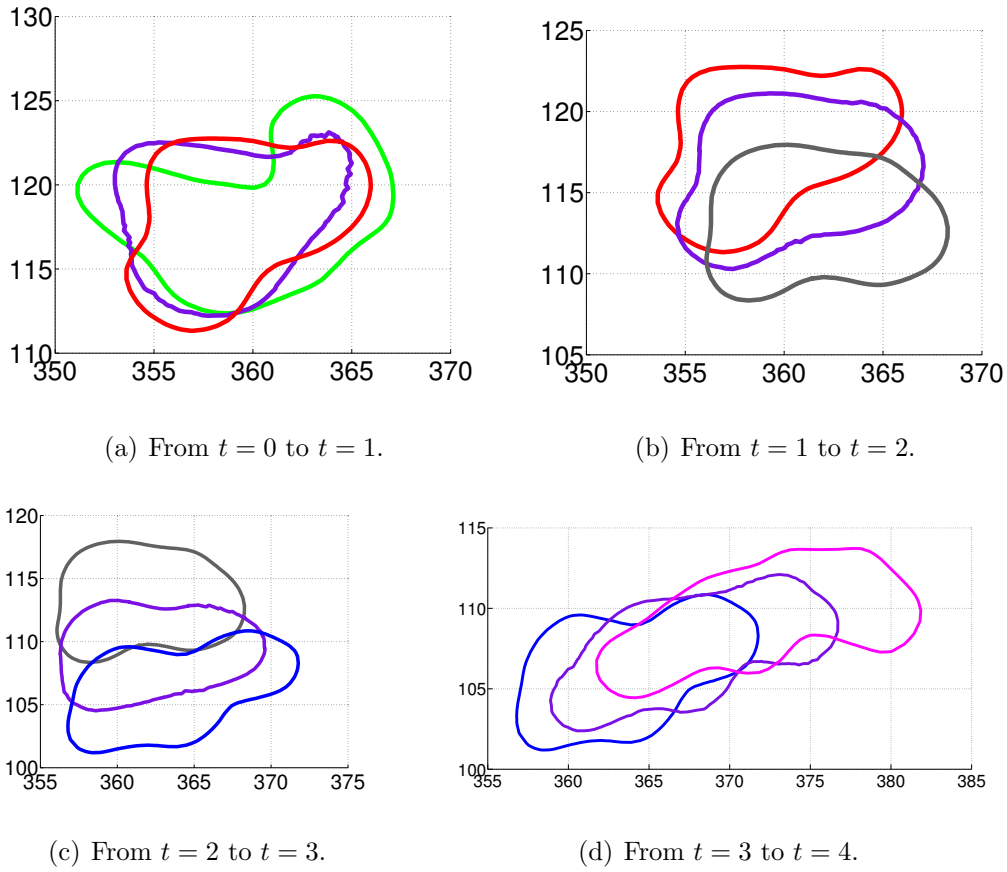
$$\mathrm{d}(\Gamma_k, \Gamma_{k+1}) = \max\left\{ \sup_{a\in\Gamma_k} \inf_{b\in\Gamma_{k+1}} \mathrm{d}(a,b), \sup_{b\in\Gamma_{k+1}} \inf_{a\in\Gamma_k} \mathrm{d}(a,b) \right\}, \tag{3.10}$$

for $k = 0, \ldots, 6$ and a, b points from Γ_k and Γ_{k+1}, respectively. Furthermore, we assumed that the mapping preserves the connectivity of the node and each polygonal line has been considered positively (anti-clockwise) oriented.

Then, once the entire mapping is defined we apply the cubic spline interpolation (3.6) on each subinterval $[t_k, t_{k+1}]$, $k = 0, \ldots, 6$ and determine the intermediate values $(\boldsymbol{x}, \boldsymbol{y})$ between the given discrete imaging data, from time $t = 0$ to time $t = 7$, every $t = 0.002$. In Figures 3.2 and 3.3 we illustrate with purple lines some intermediate positions of the neutrophil evolution though the biologically observed images.

Also, in Figure 3.4 we show the trajectories of four points of the cell boundary of the neutrophil, from time $t = 0$ to time $t = 7$.

Based on the numerical findings we are able to compute the area and the centroid (centre of mass) on each cell through the time of simulation. Also, using the trajectory of the centroids we are able to calculate some measures of cell motility such as persistence length and speed of the centroid, but these are not reported in this chapter. Later in this thesis, we will compare and contrast the interpolation method and the optimal control approach for cell tracking, and it is in that chapter that some measures will be exhibited (see Chapter 5).



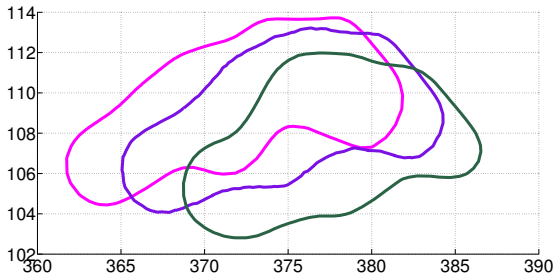
(a) From $t = 0$ to $t = 1$.

(b) From $t = 1$ to $t = 2$.

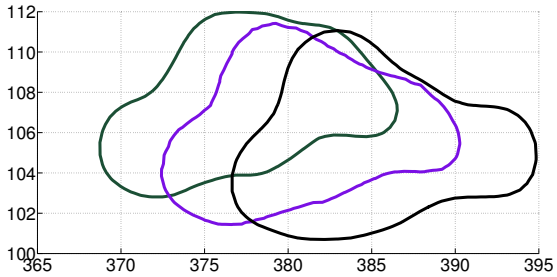
(c) From $t = 2$ to $t = 3$.

(d) From $t = 3$ to $t = 4$.

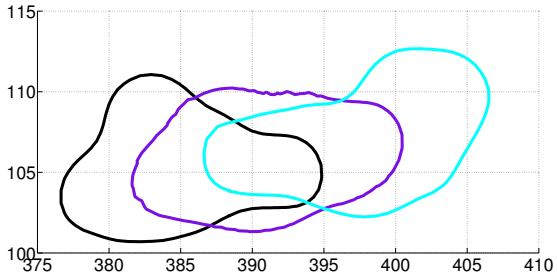
Figure 3.2: Intermediate positions of the locations of the cell membrane of the neutrophil through the biologically observed images using cubic spline interpolation. The purple polygonal line is the computed intermediate position of the cell membrane locations of the neutrophil at $t = \frac{t_{k+1}-t_k}{2}$, after the application of the cubic spline interpolation.



(a) From $t = 4$ to $t = 5$.



(b) From $t = 5$ to $t = 6$.



(c) From $t = 6$ to $t = 7$.

Figure 3.3: Figure 3.2 continued. Intermediate positions of the locations of the cell membrane of the neutrophil through the biologically observed images using cubic spline interpolation. The purple polygonal line is the computed intermediate position of the cell membrane locations of the neutrophil at $t = \frac{t_{k+1}-t_k}{2}$, after the application of the cubic spline interpolation.

3.4 Conclusion

In this chapter we presented a simple single cell tracking algorithm based on the cubic interpolation. An attractive feature of the interpolation method is that it is easy to use and can provide intermediate cell positions associated with cell tracking.

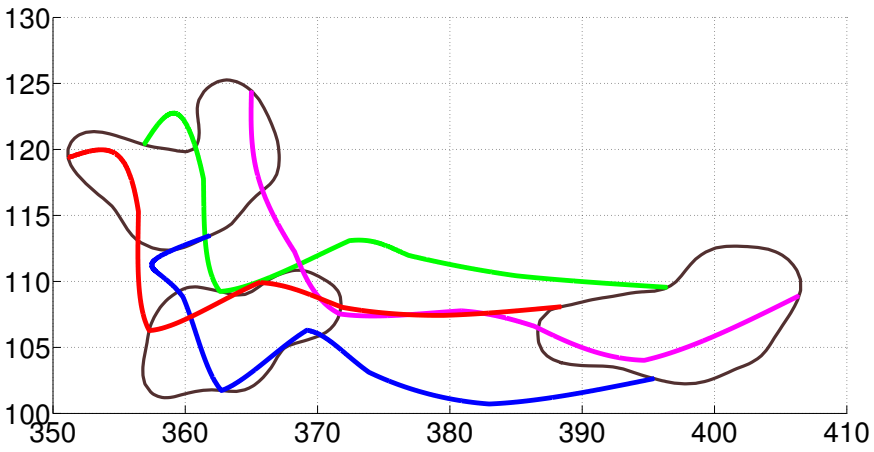


Figure 3.4: Trajectories of four points of the cell boundary of the neutrophil, from time $t = 0$ to time $t = 7$.

However, this method has several issues with low regularity being one of them. In addition, such method can generate polygonal lines in which remeshing is essential. Another drawback with the interpolation method as a cell tracking method is that there is no physical law for the evolution of the cell motion.

Here, we implemented the proposed methodology on real biological data of neutrophils as observed in zebrafish *Danio rerio larvae*. Our approach allows us to make predictions on the locations of the intermediate cell shapes between the experimental frames and to construct a continuous trajectory of the cell membrane including the cell centroid. In addition, usual measures for cell motility can be easily extracted from this approach, such as confinement ratio, area of the cell and the speed of cell centroid.

However, this framework could become a useful and easy to implement, computational tool for the experimentalists to track cells and recover trajectories from discrete image data. Consecutively, it is a simple approach to construct continuous trajectories of the cell centroid, rather than connecting linearly the cell centroids from the static imaging data. In the following chapter, we will develop an alternative approach for cell tracking where the cell tracking is based on physical laws.

Chapter 4

Optimal control

4.1 Introduction

In recent years, the rapid development in microscopy and imaging techniques has generated a huge amount of data on migrating cells both *in vivo* and *in vitro* (Pittet and Weissleder [2011]). Inferring dynamic quantities from this static data is an important task that has many applications in biology and related fields. The field of cell tracking arose from this need and is concerned with the development of methods to track and analyse dynamic cell shape changes from a series of still images captured within a time frame.

On the other hand, a major focus of current research is the derivation of mathematical models for cell migration based on physical principles, (e.g., Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). Furthermore, such models appear to show good qualitative and quantitative agreement with experimental observations of migrating cells. Despite this, very little research has focused on incorporating these mathematical modelling advances into appropriate cell tracking algorithms. Croft et al. [2013] investigated fitting parameters in models for cell motility to experimental image data sets of migrating cells where observations of both the position of the cells and the concentrations of cell-resident proteins related to motility were available.

In this chapter we present a first step towards the development of a framework for cell tracking based on novel models of cell motility. Specifically, we propose a cell tracking algorithm which can be thought of as fitting a simplified, yet physically

meaningful, model for cell migration to experimental observations and data. We focus on the setting, prevalent in cell tracking problems, where only the position of the cell at a series of discrete times is available and no further biological information is given. Here the tracking procedure we propose allows us to incorporate physically important aspects of cell migration, for example, by including volume constraints in the model for the evolution. This is motivated by the observation that, for many cells, while the surface area of the cell membrane may change significantly during migration the volume enclosed by the cell remains roughly constant (Shao et al. [2010]).

We present a mathematical model based on physical principles for the cell movement that consists of a geometric evolution equation. We then formulate an inverse problem, which takes the form of a partial differential equation constrained optimisation problem, for fitting the model to the static experimental observations. Our algorithm is inspired by Haußer et al. [2010] and Haußer et al. [2012], who use an optimal control approach to describe the shape evolution of nanoscale islands.

We reformulate our model into the phase-field framework, which appears more suitable for the problem in hand. The phase-field method is a powerful computational method that can handle complex topological changes. Phase-field modelling has been used in a broad range of applications in material science (Steinbach [2009]), fluid dynamics (Anderson et al. [1998]), biology (Du et al. [2005]) and in solving phase transition problems. In addition, the phase-field method has been applied to a wide range of applications related to cell motility. For example, Ziebert et al. [2011] presented a mathematical model within a phase-field approach, where the cell shape dynamics are coupled to a vector field describing the polarisation of the actin filament network. In addition, Marth and Voigt [2013] introduced a Helfrich type model for bending and hydrodynamic interactions of cell membrane to derive a continuous model within a phase-field approach.

The phase-field approach to interface evolution is related to problems involving phase transitions. In our problem, an auxiliary field is introduced, that distinguish the interior of the exterior of the cell (two phases). In this chapter, Ω is a bounded domain in $\mathbb{R}^2$ and $\Gamma(t)$ is the closed surface represents the moving cell membrane through Ω. Within this phase-field formulation, the moving cell membrane $\Gamma(t)$ is

represented by an evolving thin interfacial layer

$$\Gamma_\varepsilon(t) = \{\boldsymbol{x} \in \mathbb{R}^{n+1} | -1 + c_\varepsilon \leq \varphi_\varepsilon(\mathbf{x}, t) \leq 1 - c_\varepsilon\}, \tag{4.1}$$

involving a small parameter ε related to its thickness. In this approach, the surface is a thin diffuse interfacial layer of width $\mathcal{O}(\varepsilon)$ across which a phase-field variable $\varphi(\boldsymbol{x}, t) : \Omega \times (0, T) \to \mathbb{R}$ has a step transition from the bulk values $\approx +1$ or ≈ -1 on either side of the interface. The zero level set of the phase-field function approximates the surface $\Gamma(t)$. The bulk values of the phase-field function correspond to the minima of a C^2 energy function $G(\cdot) : \mathbb{R} \to \mathbb{R}$ with two equal double wells (Deckelnick et al. [2005]). An important aspect of this methodology is that does not require explicit tracking of the moving cell boundary.

In summary, the key novelty of the proposed algorithm lies in the fact that the model that drives the tracking procedure is a simplification of recently derived physically motivated models for cell motility. We model the evolution of the cell membrane as being governed by volume conserved mean curvature flow with forcing, given by

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x}, t) = (-\sigma H(\boldsymbol{x}, t) + \eta(\boldsymbol{x}, t) + \lambda_V(t))\, \boldsymbol{\nu}(\boldsymbol{x}, t) \quad \text{on } \Gamma(t), t \in (0, T], \\ \Gamma(0) = \Gamma^0, \end{cases} \tag{4.2}$$

where Γ is the closed surface that represents the cell membrane, $\boldsymbol{V}$ is the material velocity of Γ, σ is the surface tension, H the mean curvature of the surface Γ (see Definition 6 in §2.2) and $\lambda_V(t)$ is a spatially uniform force accounting for volume conservation. The forcing function η is the main driver of the directed migration and serving as a control variable for the cell-surface evolution. The equation (4.2) can be approximated by a phase-field model which is a driven Allen-Cahn equation with forcing (e.g., Evans et al. [1992])

$$\begin{cases} \partial_t \varphi(\boldsymbol{x}, t) & = \Delta\varphi(\boldsymbol{x}, t) - \frac{1}{\varepsilon^2} G'(\varphi(\boldsymbol{x}, t)) - \frac{1}{\varepsilon} \left(c_G \eta(\boldsymbol{x}, t) - \lambda_V(t)\right) \text{ in } \Omega \times (0, T], \\ \nabla\varphi \cdot \boldsymbol{\nu}_\Omega & = 0 \text{ on } \partial\Omega \times (0, T], \\ \varphi(\cdot, 0) & = \varphi^0(\cdot) \text{ in } \Omega, \end{cases} \tag{4.3}$$

where $\Omega \subset \mathbb{R}^d$ is a bulk domain with normal $\boldsymbol{\nu}_\Omega$ that contains Γ, φ^0 is a diffuse interface representation of Γ^0 and $\varepsilon > 0$ is a small parameter which governs the

width of the diffuse interface (interfacial thickness). The function G is a double well potential and c_G is scaling constant that depends on the double well potential.

Within this phase-field approach, we formulate the cell tracking problem as a partial differential equation constrained optimal control problem. We define the objective functional

$$J(\varphi, \eta) = \frac{1}{2} \int_\Omega \left(\varphi(\boldsymbol{x}, T) - \varphi_{obs}(\boldsymbol{x})\right)^2 \, \mathrm{d}\boldsymbol{x} + \frac{\theta}{2} \int_0^T \int_\Omega \eta(\boldsymbol{x}, t)^2 \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t, \tag{4.4}$$

where φ_{obs} is a diffuse interface representation of the observation Γ_{obs} of the cell position and $\theta > 0$ is a regularisation parameter; which has to be minimised for a space and time control function $\eta(\boldsymbol{x}, t)$.

The goal is to drive the cell-surface using the control function η in such way that the position of the cell membrane at time $t = T$, is close as possible to Γ_{obs}. Then, by utilising a Lagrangian framework ([Tröltzsch, 2010]), we derive the necessary first order optimality conditions and solve the system of the state and adjoint problem by using the finite element method. Finally, a gradient based update of the control is used.

The significance of this algorithm lies in the fact that it is, to our best knowledge, the first study that encompasses research at the forefront of modelling cell migration with cell tracking. Furthermore, this study has the intention to recover physically meaningful dynamic data from static imaging data sets.

The remainder of this chapter proceeds as follows. In Section 4.2 we introduce our approach to cell tracking, which may be regarded as fitting a mathematical model to experimental image data sets. We present the geometric evolution law model we seek to fit, which is a simplification of recently developed models in the literature that show good agreement with experiments (Shao et al. [2010]; Ziebert et al. [2011]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]; Shao et al. [2012]; Marth and Voigt [2013]). We finish Section 4.2 by formulating the cell tracking problem as a partial differential equation constrained optimisation problem. In Section 4.3 we present the finite element method for solving the forward and adjoint problems. In addition, we present the update scheme for the control variable. In Section 4.4 we propose an algorithm for the resolution of the partial differential equation constrained optimisation problem and we discuss some practical aspects related to the implementation. In particular we note that the proposed theoretical and

computational framework can be easily applied directly to multi-cell and raw image data sets (of sufficient quality) without segmentation. In Section 4.5 we present some classical numerical examples in order to validate and test our algorithm. In particular, the proposed numerical algorithm is applied to two-dimensional synthetic single and multi cell image data sets. Finally, in Section 4.6 we present some conclusions of our study and discuss future extensions and applications of the work.

4.2 Problem Formulation

4.2.1 Model

As mentioned above in contrast to many of the existing approaches for cell tracking, the framework we propose is based on fitting a model, derived from physical principles, for the motion of the cell to experimental image data. The general class of models to which our approach is applicable are partial differential equations based models for the motion, where the movement of the cell membrane is described by a geometric evolution law.

We model the evolution of the cell membrane as being governed by volume conserved mean curvature flow with forcing, given by

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x},t) = \left(-\sigma H(\boldsymbol{x},t) + \eta(\boldsymbol{x},t) + \lambda_V(t)\right)\boldsymbol{\nu}(\boldsymbol{x},t) \quad \text{on } \Gamma(t), t \in (0,T], \\ \Gamma(0) = \Gamma^0, \end{cases} \tag{4.5}$$

where Γ is the closed surface that represents the cell membrane, $\boldsymbol{V}$ is the material velocity of Γ, σ is the surface tension, H the mean curvature of the surface Γ (see Definition 6 in Section 2.2) and $\lambda_V(t)$ is a spatially uniform force accounting for volume conservation. Physically, this may be thought of as an interior pressure. The control function $\eta(\boldsymbol{x},t)$ enters the equation of the motion in the form of an additional driving force which acts normal on the surface. The goal is to drive the cell-surface evolution using this force in such way that the desired cell shape is obtained. The forcing function η is the main driver of the directed migration. The model we present is phenomenological and hence it is difficult to directly relate η to biophysical processes. However, as positive values of η correspond to protrusive forces and negative values of η correspond to contractile forces one interpretation

of the forcing function η is that it accounts for both protrusive forces generated by polymerisation of actin at the leading edge of the cell and contractile forces generated by the action of myosin motors at the rear of the cell.

The evolution law (4.5) is a simplification of a large class of models that arise in the modelling of cell motility which take the following form (Neilson et al. [2010]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012])

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x},t) = \Big(g_1(H(\boldsymbol{x},t)) + g_2(\boldsymbol{a}(\boldsymbol{x},t)) + \lambda_V(t) \Big) \boldsymbol{\nu}(\boldsymbol{x},t) \quad \text{on } \Gamma(t), t \in (0,T], \\ \Gamma(0) = \Gamma^0, \end{cases} \tag{4.6}$$

where g_1 models the dependence of the evolution on geometric quantities, such as resistance of the membrane to stretching which could be modelled by mean curvature terms as in (4.5). The function g_2 appearing in (4.6) captures the dependence of the evolution on a vector of bulk and/or surface resident species $\boldsymbol{a}(\boldsymbol{x},t)$. The surface resident species $\boldsymbol{a}(\boldsymbol{x},t)$ could satisfy another partial differential equation such as a surface reaction-diffusion system

$$\begin{cases} \partial^\bullet_{\boldsymbol{V}} \boldsymbol{a} + \boldsymbol{a} \left(\nabla_{\Gamma(t)} \cdot \boldsymbol{V} \right) - \boldsymbol{D} \Delta_{\Gamma(t)} \boldsymbol{a} = \boldsymbol{f}(\boldsymbol{a}) \quad \text{on } \Gamma(t), t \in (0,T], \\ \boldsymbol{a}(\cdot,0) = \boldsymbol{a}^0(\cdot) \quad \text{on } \Gamma(0), \end{cases} \tag{4.7}$$

where $\boldsymbol{a} = (a_1, \ldots, a_{n_a})^T$, n_a is the number of chemical species involved, a_i denotes the density of the ith chemical species,

$$\partial^\bullet_{\boldsymbol{V}} \boldsymbol{a} := \partial_t \boldsymbol{a} + \boldsymbol{V} \cdot \nabla \boldsymbol{a}, \tag{4.8}$$

is the material derivative with respect to the velocity $\boldsymbol{V}$, $\boldsymbol{D}$ is a diagonal matrix of positive constant diffusion coefficients and $\boldsymbol{f}(\boldsymbol{a})$ is a vector function coupling non-linear reactions. In the above, since the surface Γ is closed, no boundary conditions are required.

Remark 9. *The model system (4.7) is an extension of the model (2.13). The only difference is that (4.7) is posed on an evolving surface while (2.13) is posed on a stationary surface.*

Models of the form (4.6)-(4.7) have been used successfully to model cell motility in Neilson et al. [2010], Neilson et al. [2011a], Neilson et al. [2011b], Elliott et al.

[2012] while models coupling evolution laws of the form (4.6) to bulk partial differential equations (i.e., equations posed in the cell interior) have been considered in Shao et al. [2012], Marth and Voigt [2013], Ziebert et al. [2011].

Despite its simplicity the evolution law (4.5) may be regarded as a prototype of more complex models of cell motility of the form (4.6)-(4.7). The geometric evolution equation (4.6) is often the most challenging component of the model to solve numerically. Developing an understanding of how to construct cell tracking algorithms by assuming a geometric evolution law based model for the motion is an important first step towards developing tracking algorithms based on more realistic physical models.

In many applications it is also the case that the only information available from the data is the position of the cell membrane and no adequate model for the biochemistry of the motility related species involved. Without any knowledge of the relevant biochemistry it is difficult to identify which motility related species should influence the evolution let alone propose how the evolution depends on their distribution (i.e., a g_2 in (4.6)) or a model for the species dynamics (i.e., an equation such as (4.7)). Nevertheless, one may still wish to extract dynamic quantities from static image data sets. Therefore, it may be reasonable to consider the evolution law (4.5) as a stand alone model for the motion as at least the mechanical aspects of the membrane evolution are accounted for through a physical model derived from basic physical principles.

4.2.2 An optimal control approach to cell tracking

The cell tracking approach we consider in this chapter corresponds to the following problem.

Problem 1 (Cell tracking)**.** *Given an initial cell membrane position* Γ^0 *and an observation of the cell position* Γ_{obs}*, find a space-time distributed forcing* η *such that the evolution of the cell membrane,* $\Gamma(t), t \in [0,T]$ *satisfies (4.5) with* $\Gamma(0) = \Gamma^0$ *and* $\Gamma(T)$ *the position of the cell membrane at time* $t = T$*, is close to* Γ_{obs}*.*

As the volume enclosed by the cell may vary over the images it is inappropriate to enforce conservation of a constant volume. Instead we enforce, with the help of

a Lagrange multiplier $\lambda(t)$, that the volume enclosed by the cell is given by

$$\widetilde{V}(t) = V^0 + \frac{t}{T}(V_{obs} - V^0),$$

i.e. that the volume of the cell is a time-dependent linear interpolant of the volumes of the data.

Problem 1 is an optimal control of a free moving boundary problem, where the free moving boundary problem is that of forced mean curvature flow and the control variable is the space time distributed forcing. We wish to exploit this fact and to this end we consider the phase-field approximation of (4.5), given by the Allen-Cahn equation with forcing;

$$\begin{cases} \partial_t \varphi(\boldsymbol{x},t) & = \Delta\varphi(\boldsymbol{x},t) - \frac{1}{\varepsilon^2} G'(\varphi(\boldsymbol{x},t)) - \frac{1}{\varepsilon}\left(c_G \eta(\boldsymbol{x},t) - \lambda(t)\right) \text{ in } \Omega \times (0,T], \\ \nabla\varphi \cdot \boldsymbol{\nu}_\Omega & = 0 \text{ on } \partial\Omega \times (0,T], \\ \varphi(\cdot,0) & = \varphi^0(\cdot) \text{ in } \Omega, \end{cases} \tag{4.9}$$

where $\Omega \subset \mathbb{R}^d$ is a bulk domain with normal $\boldsymbol{\nu}_\Omega$ that contains Γ, φ^0 is a diffuse interface representation of Γ^0 and $\varepsilon > 0$ is a small parameter which governs the width of the diffuse interface (interfacial thickness). For details on the asymptotic analysis of (4.9) and the convergence (as $\varepsilon \to 0$) to a solution of (4.5) we refer the reader, for example, to Chen [1992]; Blowey and Elliott [1993]; Bellettini and Paolini [1996]; Brassel and Bretin [2011] and the references therein. The function G appearing in (4.9) is a double well potential, for example the quartic potential

$$G(\varphi) = \frac{1}{4}\left(\varphi^2 - 1\right)^2 \tag{4.10}$$

which has minima at ± 1. The constant

$$c_G = \frac{1}{\sqrt{2}} \int_{-1}^{1} G(r)^{1/2} \, \mathrm{d}r \tag{4.11}$$

appearing in (4.9) is a scaling constant that depends on the double well potential.

Remark 10. *The Haussdorff distance could be employed in order to measure and compare the positions of a computed curve and a given observation (desired position). However, this methodology gives rise to smoothness issues that do not allow the formulation of the adjoint problem. For this reason, phase-field framework appears more suitable for our problem.*

Remark 11 (Choice of the potential)**.** *We note that our approach to the optimal control problem involving the formulation of the adjoint problem appears to require a smooth potential G (c.f.,* (4.10)*). The formulation of the adjoint problem is to our best knowledge an open problem for other widely used, but non smooth or unbounded, potentials such as the obstacle or logarithmic potential.*

We enforce the time-dependent volume constraint following the approach of (Blowey and Elliott [1993]). Specifically our diffuse interface formulation of the constraint on the enclosed volume is given by a constraint on $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$, where $[a]_+ = max(a,0)$. We define the linear interpolant of $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$ of the initial and target diffuse interface data M_φ by

$$M_\varphi(t) := \int_\Omega [\varphi^0]_+ + \frac{t}{T}\left([\varphi_{obs}]_+ - [\varphi^0]_+\right) \mathrm{d}\boldsymbol{x}, \tag{4.12}$$

and determine $\lambda(t)$ in (4.9) such that

$$M_\varphi(t) = \int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}. \tag{4.13}$$

We have used λ (rather than λ_V) for the Lagrange multiplier in (4.9) to reflect the fact that our constraint is on $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$. However, we shall refer to this constraint as a volume constraint in order to highlight the physical feature the constraint is intended to model. We also investigated an alternative approach to enforcing the volume constraint via penalising deviations from a target volume following Du et al. [2006]. In all our numerical tests this strategy proved less robust than the volume constraint proposed above.

Remark 12 (Volume constraint)**.** *Here, we have to point out that our diffuse interface formulation of the constraint is actually given by a constraint on the mass rather than the volume. We define the linear interpolant of the mass of the initial and target diffuse interface data by M_φ. This formulation has been taken from Blowey and Elliott [1993], where this method has been applied for a double obstacle problem and as $\varepsilon \to 0$ conservation of mass is yields volume conservation.*

To formulate the cell tracking problem as a partial differential equation constrained optimal control problem we define the objective functional we shall seek to minimise as follows

$$J(\varphi,\eta) = \frac{1}{2}\int_\Omega \left(\varphi(\boldsymbol{x},T) - \varphi_{obs}(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x} + \frac{\theta}{2}\int_0^T \int_\Omega \eta(\boldsymbol{x},t)^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t, \tag{4.14}$$

where φ_{obs} is a diffuse interface representation of the observation Γ_{obs} and $\theta > 0$ is a regularisation parameter. The first term on the right of (4.14) is the so-called fidelity term that measures the distance between the solution to the model and the target data and the second term is the regularisation which is necessary to ensure a well-posed problem (for example see [Tröltzsch, 2010]).

Our optimal control approach to the cell tracking problem may now be stated as the following minimisation problem.

Problem 2 (Optimal control problem)**.** *Given an initial diffuse interface representation of the cell membrane position φ^0 and an observation of the cell membrane position φ_{obs}, find a space-time distributed forcing $\eta^* : \Omega \times [0,T] \to \mathbb{R}$ such that with φ a solution of (4.9) with initial condition $\varphi(\cdot,0) = \varphi^0$, the forcing η^* solves the minimisation problem*

$$\min_{\eta} J(\varphi,\eta), \text{ where } J \text{ is given by (4.14).} \tag{4.15}$$

Remark 13 (Well-posedness of the problem)**.** *The study of the well-posedness of the optimal control problem 2 is a very challenging mathematical task and it is out of the scope of this thesis. However, we refer to Blank et al. [2013] where the well-posedness of similar problems have been studied.*

Remark 14 (Local minima)**.** *For the optimal control problem 2, there can exist more than one local minima. In Section 4.5.3 we present a numerical example, to illustrate the effect that the choice of the initial guess for the control η has on the solution of the problem.*

4.2.3 Optimality conditions

To apply the theory of optimal control of semilinear partial differential equation for the solution of the tracking problem, we briefly outline the derivation of the optimality conditions. For further details we refer to Hinze et al. [2009] and Tröltzsch [2010]. Introducing the Lagrange multiplier (adjoint state) p, we define the Lagrangian functional

$$\begin{aligned}\mathcal{L}(\varphi,\eta,p) = J(\varphi,\eta) - \int_0^T \int_\Omega \Big(&\partial_t\varphi(\boldsymbol{x},t) - \Delta\varphi(\boldsymbol{x},t) \\ &+ \frac{1}{\varepsilon^2} G'(\varphi(\boldsymbol{x},t)) + \frac{1}{\varepsilon}\big(c_G\eta(\boldsymbol{x},t) - \lambda(t)\big)\Big) p(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t.\end{aligned} \tag{4.16}$$

Requiring stationarity of the Lagrangian with respect to the adjoint state yields the state equation (4.9) and requiring stationarity of the Lagrangian, at the optimal control η^* and associated optimal state φ^*, with respect to the state and the control, yields the (formal) necessary first order optimality conditions (Tröltzsch [2010])

$$D_\varphi \mathcal{L}(\varphi^*, \eta^*, p)\varphi = 0, \quad \forall \varphi \in H^1(\Omega) : \varphi(\boldsymbol{x}, 0) = 0, \tag{4.17}$$

$$D_\eta \mathcal{L}(\varphi^*, \eta^*, p)\eta = 0, \quad \forall \eta, \tag{4.18}$$

where D denotes the functional derivative. The Lagrangian functional, using integration by parts and the boundary conditions of (4.9), can be calculated explicitly as

$$\begin{aligned}
\mathcal{L}(\varphi^*, \eta^*, p) &= \frac{1}{2}\int_\Omega (\varphi^* - \varphi_{obs})^2 \,\mathrm{d}\boldsymbol{x} + \frac{\theta}{2}\int_0^T \int_\Omega \eta^{*2} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \int_0^T \int_\Omega \big(p\partial_t \varphi^* \\
&\quad - p\Delta\varphi^* + \frac{1}{\varepsilon^2} pG'(\varphi^*)) + \frac{1}{\varepsilon}\big(c_G\eta^* - \lambda(t)\big)p\big) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&= \frac{1}{2}\int_\Omega (\varphi^* - \varphi_{obs})^2 \,\mathrm{d}\boldsymbol{x} + \frac{\theta}{2}\int_0^T \int_\Omega \eta^{*2} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \int_0^T \int_\Omega \big(p\partial_t \varphi^* \\
&\quad + \nabla\varphi^* \cdot \nabla p + \frac{1}{\varepsilon^2} pG'(\varphi^*) + \frac{1}{\varepsilon}\big(c_G\eta^* - \lambda(t)\big)p\big) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\quad + \int_0^T \int_{\partial\Omega} p\nabla\varphi^* \cdot \boldsymbol{\nu}_\Omega \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&= \frac{1}{2}\int_\Omega (\varphi^* - \varphi_{obs})^2 \,\mathrm{d}\boldsymbol{x} + \frac{\theta}{2}\int_0^T \int_\Omega \eta^{*2} \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \int_0^T \int_\Omega \big(p\partial_t \varphi^* \\
&\quad + \varphi^*\Delta p + \frac{1}{\varepsilon^2} pG'(\varphi^*) + \frac{1}{\varepsilon}\big(c_G\eta^* - \lambda(t)\big)p\big) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\quad - \int_0^T \int_{\partial\Omega} \varphi^*\nabla p \cdot \boldsymbol{\nu}_\Omega \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t.
\end{aligned}$$

Remark 15. *We note that the above calculation is a formal procedure, as the adjoint state p might not be of sufficient regularity, in general.*

Then, we have

$$\begin{aligned}
D_\varphi \mathcal{L}(\varphi^*, \eta^*, p)\varphi &= \int_\Omega (\varphi^* - \varphi_{obs})\varphi \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \int_\Omega [p\varphi]_0^T \,\mathrm{d}\boldsymbol{x} \\
&\quad - \int_0^T \int_\Omega \big(- \partial_t p - \Delta p + \frac{1}{\varepsilon^2} G''(\varphi^*)p\big)\varphi \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t = 0 \\
&\quad - \int_0^T \int_{\partial\Omega} \varphi\nabla p \cdot \boldsymbol{\nu}_\Omega \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t.
\end{aligned}$$

Therefore, imposing Neumann boundary conditions for the adjoint state p, i.e. $\nabla p \cdot \boldsymbol{\nu}_\Omega = 0$, and using the condition $\varphi(\boldsymbol{x}, 0) = 0$ from (4.17) and pick a $\varphi(\boldsymbol{x}, t)$ with

support only at the final time T, yields the adjoint problem

$$\begin{cases} \partial_t p(\boldsymbol{x},t) &= -\Delta p(\boldsymbol{x},t) + \frac{1}{\varepsilon^2} G''(\varphi(\boldsymbol{x},t)) p(\boldsymbol{x},t) \quad \text{in } \Omega \times (0,T], \\ \nabla p \cdot \boldsymbol{\nu}_\Omega &= 0 \quad \text{on } \partial\Omega \times (0,T], \\ p(\boldsymbol{x},T) &= \varphi(\boldsymbol{x},T) - \varphi_{obs}(\boldsymbol{x}) \quad \text{in } \Omega. \end{cases} \tag{4.19}$$

Note that equation (4.19) is posed backwards in time and hence is equipped with terminal conditions.

Similarly, from condition (4.18) together with the Riesz representation theorem yields the optimality condition (c.f., Tröltzsch [2010])

$$D_\eta \mathcal{L}(\varphi^*, \eta^*, p)\eta = \int_0^T \int_\Omega \left(\theta \eta^* + \frac{c_G}{\varepsilon} p\right) \eta = 0, \tag{4.20}$$

and since η can be arbitrarily picked we conclude that

$$\theta \eta^* + \frac{c_G}{\varepsilon} p = 0. \tag{4.21}$$

In practice the forward (4.9) and adjoint equations (4.19) must be solved numerically and we employ the finite element method.

4.3 Numerical approximation

The forward (4.9) and adjoint equations (4.19) must be solved numerically. For this reason, we employ the finite element method to approximate the solutions corresponding to the forward and the adjoint partial differential equations (Deckelnick et al. [2005]). The basic idea is that for each step of the loop in our algorithm, we first solve the state equation (4.9) with a given control, then solve the adjoint equation (4.19) with the computed states and then update the control using the optimality condition (4.21). For this initial study we employ a simple gradient based update of the control (Tröltzsch [2010]).

Below we outline how the numerical scheme is derived for the state (4.9) and adjoint equations (4.19). In addition, we present the update scheme that is used for the control function $\eta(\boldsymbol{x},t)$.

4.3.1 Discretisation of the forward equation

We introduce the variational form for the forward problem (4.9) defined as follows. Find $(\varphi, \lambda) \in L^2([0,T]; H^1(\Omega)) \times L^2(0,T)$ such that

$$\int_\Omega \partial_t \varphi \psi \, \mathrm{d}\boldsymbol{x} + \int_\Omega \nabla\varphi \cdot \nabla\psi \, \mathrm{d}\boldsymbol{x} = \frac{1}{\varepsilon}\int_\Omega (c_G\eta - \lambda)\psi \, \mathrm{d}\boldsymbol{x} - \frac{1}{\varepsilon^2}\int_\Omega G'(\varphi)\psi \, \mathrm{d}\boldsymbol{x}, \quad \forall \psi \in H^1(\Omega).$$

Let $\mathcal{T}$ be a decomposition of Ω into simplexes S (for simplicity we assume Ω is a polygonal domain), i.e. $\Omega = \bigcup_{k\in\mathcal{T}} S$. Furthermore, we define $h = max_{S\in\mathcal{T}}\{\text{diam } S\}$ the maximal diameter of $\mathcal{T}$ and the set $\mathcal{J}$ to be the set of nodes of $\mathcal{T}$. We define the finite element space by

$$\mathcal{V} := \Big\{\psi_h \in H^1(\Omega) \cap C^0(\Omega) : \psi_h|_k \in \mathbb{P}^1 \quad \forall k \in \mathcal{T}\Big\}. \tag{4.22}$$

In addition let $\{\chi_j\}_{j\in\mathcal{J}}$ to be the standard nodal basis functions of $\mathcal{V}$. Then by $\varphi_j(t)$ for $j = 1, \ldots, \mathcal{J}$ we denote the coefficients of the basis representation. Thus, for the discrete solution $\varphi_h \in \mathcal{V}$ we have

$$\varphi_h(\boldsymbol{x}, t) = \sum_{j\in\mathcal{J}} \varphi_j(t)\chi_j(\boldsymbol{x}). \tag{4.23}$$

For the time discretisation we employ an implicit-explicit method where the diffusive term is treated implicitly and the nonlinear reaction terms explicitly. For more details about the implicit-explicit discretisation schemes we refer to and Ascher et al. [1995] and Madzvamuse [2006]. We discretise the time $[0,T]$ interval into a finite number of sub-intervals. Introducing the shorthand for a time discrete sequence $f^n := f(t^n)$ and a uniform timestep τ with $T = M\tau, M \in \mathbb{N}$, the fully discrete scheme reads, for $n = 0, \ldots, M-1$, given $\varphi_h^n, \eta_h^n \in \mathcal{V}$, find $(\varphi_h^{n+1}, \lambda^{n+1}) \in \mathcal{V} \times \mathbb{R}$ such that

$$\begin{aligned}\frac{1}{\tau}\int_\Omega (\varphi_h^{n+1} - \varphi_h^n)\psi_h \, \mathrm{d}\boldsymbol{x} + \int_\Omega \nabla\varphi_h^{n+1} \cdot \nabla\psi_h \, \mathrm{d}\boldsymbol{x} \\ = \frac{1}{\varepsilon}\int_\Omega (c_G\eta_h^n - \lambda^{n+1})\psi_h \, \mathrm{d}\boldsymbol{x} - \frac{1}{\varepsilon^2}\int_\Omega \Lambda^h(G'(\varphi_h^n))\psi_h \, \mathrm{d}\boldsymbol{x}, \quad \forall \psi_h \in \mathcal{V},\end{aligned}$$

where $\Lambda^h : C^0(\Omega) \to \mathcal{V}$ denotes the Lagrange interpolant.

We solve the above problem using the iterative technique introduced and studied by Blowey and Elliott [1993], that uses a bisection method for the Lagrange multiplier. In particular we seek an iterative sequence $\{\varphi_h^{n+1,l}, \lambda^{n+1,l}\}_{l\geq 1}$ where $\varphi_h^{n+1,l}$

solves

$$\frac{1}{\tau}\int_\Omega(\varphi_h^{n+1,l}-\varphi_h^n)\psi_h\,\mathrm{d}\boldsymbol{x}+\int_\Omega\nabla\varphi_h^{n+1,l}\cdot\nabla\psi_h\,\mathrm{d}\boldsymbol{x}= \\ \frac{1}{\varepsilon}\int_\Omega(c_G\eta_h^n-\lambda^{n+1,l})\psi_h\,\mathrm{d}\boldsymbol{x}-\frac{1}{\varepsilon^2}\int_\Omega\Lambda^h(G'(\varphi_h^n))\psi_h\,\mathrm{d}\boldsymbol{x}, \tag{4.24}$$

for all $\psi_h\in\mathcal{V}$. We take $\lambda^{n+1,1}=-\frac{2\varepsilon}{\tau}+1$, $\lambda^{n+1,2}=\frac{2\varepsilon}{\tau}-1$ and compute $\{\lambda^{n+1,l+1}\}_{l\geq 2}$ from

$$\lambda^{n+1,l+1}=\lambda^{n+1,l}+\frac{\left(\lambda^{n+1,l}-\lambda^{n+1,l-1}\right)\left(M_\varphi^{n+1}-\int_\Omega[\varphi_h^{n+1,l}]_+\,\mathrm{d}\boldsymbol{x}\right)}{\left(\int_\Omega[\varphi_h^{n+1,l}]_+\,\mathrm{d}\boldsymbol{x}-\int_\Omega[\varphi_h^{n+1,l-1}]_+\,\mathrm{d}\boldsymbol{x}\right)},$$

where we recall (4.12)

$$M_\varphi^{n+1}:=\int_\Omega[\varphi_h^0]_+ +\frac{(n+1)\tau}{T}\left([\varphi_{obs}]_+-[\varphi_h^0]_+\right)\,\mathrm{d}\boldsymbol{x}.$$

We deem this iteration to have converged when $|\lambda^{n+1,l+1}-\lambda^{n+1,l}|<\mathrm{tol}_\lambda$. Also, we set l to be an inner iterative loop for the solution of the Lagrange multiplier and the maximum number of iterations to be $l_{max}=2000$.

Finally, the equation (4.24) can be written in matrix form as

$$\left(\frac{1}{\tau}\mathcal{M}+\mathcal{S}\right)\boldsymbol{\Phi}^{n+1,l}=\left(\frac{1}{\tau}\mathcal{M}+\mathcal{K}\right)\boldsymbol{\Phi}^n+\boldsymbol{\mathcal{F}}, \tag{4.25}$$

where $\mathcal{M}$ and $\mathcal{S}$ are the mass and the stiffness matrices, respectively, with entries

$$(\mathcal{M})_{ij}=\int_\Omega\chi_i\chi_j\,\mathrm{d}\boldsymbol{x}\quad\text{and}\quad(\mathcal{S})_{ij}=\int_\Omega\nabla\chi_i\cdot\nabla\chi_j\,\mathrm{d}\boldsymbol{x}. \tag{4.26}$$

The entries of the matrix $\mathcal{K}$ and the vector $\boldsymbol{\mathcal{F}}$ are defined by

$$(\mathcal{K})_{ij}=-\frac{1}{\varepsilon^2}\int_\Omega((\Phi^n)^2-1)\chi_i\chi_j\,\mathrm{d}\boldsymbol{x}\quad\text{and}\quad(\boldsymbol{\mathcal{F}})_j=\frac{1}{\varepsilon}(c_G\eta_h^n-\lambda^{n+1,l})\int_\Omega\chi_j\,\mathrm{d}\boldsymbol{x}, \tag{4.27}$$

respectively.

The discretisation of the forward problem without the volume constraint is trivial, by taking $\lambda=0$. We eschew the details.

4.3.2 Discretisation of the adjoint equation

Likewise, for the adjoint problem (4.19) the weak formulation reads: Find $p_h \in L^2([0,T]; H^1(\Omega))$ such that

$$\begin{aligned}\int_\Omega \partial_t p_h \omega_h \,\mathrm{d}\boldsymbol{x} &= -\int_\Omega \Delta p_h \omega_h \,\mathrm{d}\boldsymbol{x} + \frac{1}{\varepsilon^2}\int_\Omega G''(\varphi_h)\omega_h \,\mathrm{d}\boldsymbol{x} \\ &= -\left(\int_\Omega \nabla p_h \cdot \nabla \omega_h \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \omega \nabla p_h \cdot \boldsymbol{\nu}_\Omega \,\mathrm{d}\boldsymbol{x}\right) + \frac{1}{\varepsilon^2}\int_\Omega G''(\varphi) p_h \omega_h \,\mathrm{d}\boldsymbol{x} \\ &= -\int_\Omega \nabla p_h \cdot \nabla \omega_h \,\mathrm{d}\boldsymbol{x} + \frac{1}{\varepsilon^2}\int_\Omega G''(\varphi_h) p_h \omega_h \,\mathrm{d}\boldsymbol{x}, \quad \forall \omega_h \in H^1(\Omega).\end{aligned}$$

Similarly, to the forward equation, we obtain the following fully discrete approximation (in matrix form) of the adjoint equation (4.19)

$$\left(\frac{1}{\tau}\mathcal{M} - \mathcal{W} + \mathcal{S}\right)\boldsymbol{P}^n = \frac{1}{\tau}\mathcal{M}\boldsymbol{P}^{n+1}, \tag{4.28}$$

where the mass and the stiffness matrices are defined as in (4.26). The matrix $\mathcal{W}$ has entries

$$\mathcal{W}_{ij} = \frac{1}{\varepsilon^2}\int (3(\Phi^n)^2 - 1)\chi_i \chi_j \,\mathrm{d}\boldsymbol{x}.$$

The values of Φ^n are the solutions of the forward problem (4.25). Since the adjoint equation must be solved backward in time, the marching direction in time for the forward and the adjoint equation is opposite to each other. Therefore, the values of all the Φ^n are coupled at all time levels with those of all P^n.

The discretisation of the backward problem without the volume constraint is the same as above, while the parameter $\lambda(t)$ does not enter the adjoint equation (4.19).

4.3.3 Update scheme for the control variable $\eta(\boldsymbol{x}, t)$

We use the optimality conditions to construct an iterative optimisation loop to solve the optimal control problem, Problem 2. As we mentioned earlier, for this initial study, we employ a simple gradient based update of the control (Tröltzsch [2010]).

Let k be the iteration number of the optimisation loop. Given η^k and p^{k+1} we compute the updated control η^{k+1} via steepest descent. That is we choose as an update direction the negative gradient, the formula for the update of the control is

$$\eta^{k+1}(\boldsymbol{x}, t) = \eta^k(\boldsymbol{x}, t) - \alpha\Big(\theta\eta^k(\boldsymbol{x}, t) + \frac{c_G}{\varepsilon}p^{k+1}(\boldsymbol{x}, t)\Big), \quad (\boldsymbol{x}, t) \in \Omega \times [0, T), \tag{4.29}$$

where α is a step size.

Remark 16. *There are various possible strategies (e.g., bisection method, Armijo's rule) in the literature, for the control of the step size α. For simplicity within this thesis we take a constant step size. We refer to Tröltzsch [2010] for an overview of such algorithms.*

4.4 Cell tracking algorithm, practical considerations and implementation

The termination criteria for the cell tracking algorithm are if either the absolute value of the objective functional J is less than a given tolerance tol_J or the update in the control is less than a given tolerance, i.e., if $||\alpha(\theta\eta^k + \frac{c_G}{\varepsilon}p)||_{L_2(\Omega\times[0,T))} < tol_\eta$ or if a maximum number of iterations K_{max} is reached or if the change in the solution through successive time steps is small enough i.e., if $||\varphi^{k+1} - \varphi^k||_{L_2(\Omega\times[0,T))} < tol_\varphi$.

Thus, the cell tracking algorithm we propose may now be stated in pseudocode as follows:

Require: Data: φ_h^0 and $(\varphi_{obs})_h$ the initial and target (discrete) diffuse interface data.

Numerical parameters: $T > 0$ end-time and $M > 0$ number of timesteps.

Optimisation parameters: K_{max}, θ, α and the tolerances $tol_J, tol_\eta, tol_\varphi, tol_\lambda$ (see Table 4.2).

Initial guess for the control: Given $(\eta_h)^0 := (\eta_h^i)^0 \in, i = 0, \ldots, M$.

Set $k := 0$

while $\Big(||\alpha(\theta(\eta_h)^k + \frac{c_G}{\varepsilon}p_h^{k+1})||_{L_2(\Omega\times[0,T))} > tol_\eta$, $||\varphi^{k+1} - \varphi^k||_{L_2(\Omega\times[0,T))} > tol_\varphi$, $J > tol_J$ *and* $k < K_{max}\Big)$ ***do***

- *Solve the state equation (4.24) for $\{(\varphi_h^i)^{k+1}, (\lambda^i)^{k+1}\}, i = 1, \ldots, M$, with $(\eta_h^i)^k$ and initial data $(\varphi_h^0)^{k+1} = \varphi_h^0$.*

- *Solve the adjoint equation (4.28) for $(p_h^i)^{k+1}, i = M-1, \ldots, 0$, with computed $(\varphi_h^i)^{k+1}$ and with terminal data $(p_h^M)^{k+1} = (\varphi_h^M)^{k+1} - (\varphi_{obs})_h$.*

- *Update control $(\eta_h^i)^{k+1} = (\eta_h^i)^k - \alpha(\theta(\eta_h^i)^k + \frac{c_G}{\varepsilon}(p_h^i)^{k+1}) \quad , i = 0, \ldots, M$.*

- *Compute J according to (4.14).*

- $k := k + 1$.

end while

Remark 17. *(Choice of regularisation parameter θ). The value of the regularisation parameter θ is of vital importance in the optimisation algorithm as it is associated with the well-posedness of the partial differential equation and the convergence to the optimal solution. In our case, the parameter value θ was determined by trial-and-error and the best value chosen is $\theta = 0.01$.*

Remark 18. *(Choice of step size α). In addition, a crucial step in our algorithm is the selection of the step size in the steepest descent method that we employ for the update of the control variable $\eta(\boldsymbol{x}, t)$. A careful selection of the step size is important since it affects the convergence of the algorithm (Tröltzsch [2010]). If it is too large the solution of the algorithm will diverge, whereas if it is too small it will take a long time to converge. For simplicity in this thesis we take a constant step size. Thus, by trial-and-error we found that a "good" value is $\alpha = 0.01$.*

Remark 19. *As we mentioned earlier, the adjoint equation must be solved backward in time using the solutions from the forward problem. This procedure makes our algorithm computationally expensive since it requires high memory usage to maintain the stored information. Thus, in general, these kind of algorithms in most of the cases are not possible to be executed in personal computers or laptops but only on high performance clusters.*

Segmentation and image data

An important aspect of any cell tracking algorithm is its ability to extract suitable data from the experimental image data set. In many cases the experimental image data set consists of grayscale observations with the intensity (brightness) indicating whether a point is in the interior or the exterior of a cell, i.e., points inside the cell appear bright for example and points outside appear dark. For many tracking algorithms this intensity data is then post processed via a segmentation algorithm (e.g., active contour methods (Chan and Vese [2001]; Dormann et al. [2002])) to yield sharp interface representations of the cell membrane.

Assuming a sharp interface representation of the cell membrane is available, diffuse interface representations may be easily initialised (Croft et al. [2013]). We

note, however, that the raw intensity data produced by many imaging procedures may already be close to a diffuse interface representation of the cell. This is typically the case when the data is relatively free of noise and the contrast between the cell and the background is high. In this case one may wish to exploit this fact in the algorithm and work with the raw image data set itself (or a post processed e.g., thresholded version), thus circumventing the extra error induced by segmentation.

Observations at multiple points in time

For clarity of exposition we focus on the case of fitting to a single observation. The approach generalises straightforwardly to multiple observations. The first term in (4.14) is simply replaced by a sum over the different times. The observations are taken as the difference between the solution (at the appropriate time) and the target data.

Multiple cells and matching problems

As mentioned above a major focus of many cell tracking algorithms is to track multiple cells in the same image and the resolution of the so called matching problem. Our approach can be applied to multi-cell image data. Here φ^0 and φ_{obs} would be diffuse interface representations of the multi-cell image data set and the diffuse interfaces would consist of multiple disjoint phases. The next step of the approach remains unchanged and the matching problem is solved implicitly in the computation of the optimal control.

There are, however, multiple practical issues which arise in this setting related to the separation between distinct cells. This affects the choice of ε, and the fact that the evolution law (4.9) allows changes in the topology of the phases. Related to the application in cell tracking, this may lead to cell splitting, the annihilation of a phase (which would correspond to the disappearance of a cell) or the nucleation of a phase (i.e., the spontaneous appearance of a cell) (Bray [2001]).

4.5 Numerical examples

4.5.1 Validation of our methodology

In this section a number of experiments are presented which illustrate the validation of our methodology. In the following experiments we consider a square domain $\Omega = [0,4] \times [0,4]$ with a triangulation of 8321 degrees of freedom (i.e. number of nodes).

In order to validate the optimal control algorithm we start by checking initially the numerical solutions that correspond to the forward problem. A typical benchmark problem for an interface which evolves by its mean curvature is the shrinking circle in $\mathbb{R}^2$. Initially, we consider the unit circle that moves according to (4.5). Taking the control variable $\eta(\boldsymbol{x}, t) = 0$ and $\lambda_V = 0$, this leads to the Allen-Cahn equation

$$\begin{cases} \varepsilon \partial_t \varphi(\boldsymbol{x}, t) & = \varepsilon \Delta \varphi(\boldsymbol{x}, t) - \frac{1}{\varepsilon} G'(\varphi(\boldsymbol{x}, t)) \text{ in } \Omega \times (0, T], \\ \nabla \varphi \cdot \boldsymbol{\nu}_\Omega & = 0 \text{ on } \partial\Omega \times (0, T], \\ \varphi(\boldsymbol{x}, 0) & = x_1^2 + x_2^2 - 1 \text{ in } \Omega. \end{cases} \tag{4.30}$$

The rest of the parameter values used are presented in Table 4.1.

End time (T)	timestep (τ)	ε	λ_V
1	1×10^{-3}	0.1	0

Table 4.1: Parameters used for the forward simulations for the examples with the shrinking and the stationary circle.

It can be shown (Deckelnick et al. [2005]), that the circle shrinks into a point in finite time (see Figure 4.5.1). The computed radius of the zero level set at time t has radius

$$R(t) = \sqrt{R_0^2 - 2t}, \quad 0 \leq t \leq \frac{R_0^2}{2},$$

where R_0 is the initial radius (see Figure 4.2). Figure 4.5.1 shows the computed phase-field representation of the solution of the Allen-Cahn equation (4.30) at different time steps. We clearly observe that the initial circle shrinks to a point after 500 iterations and disappears afterwards.

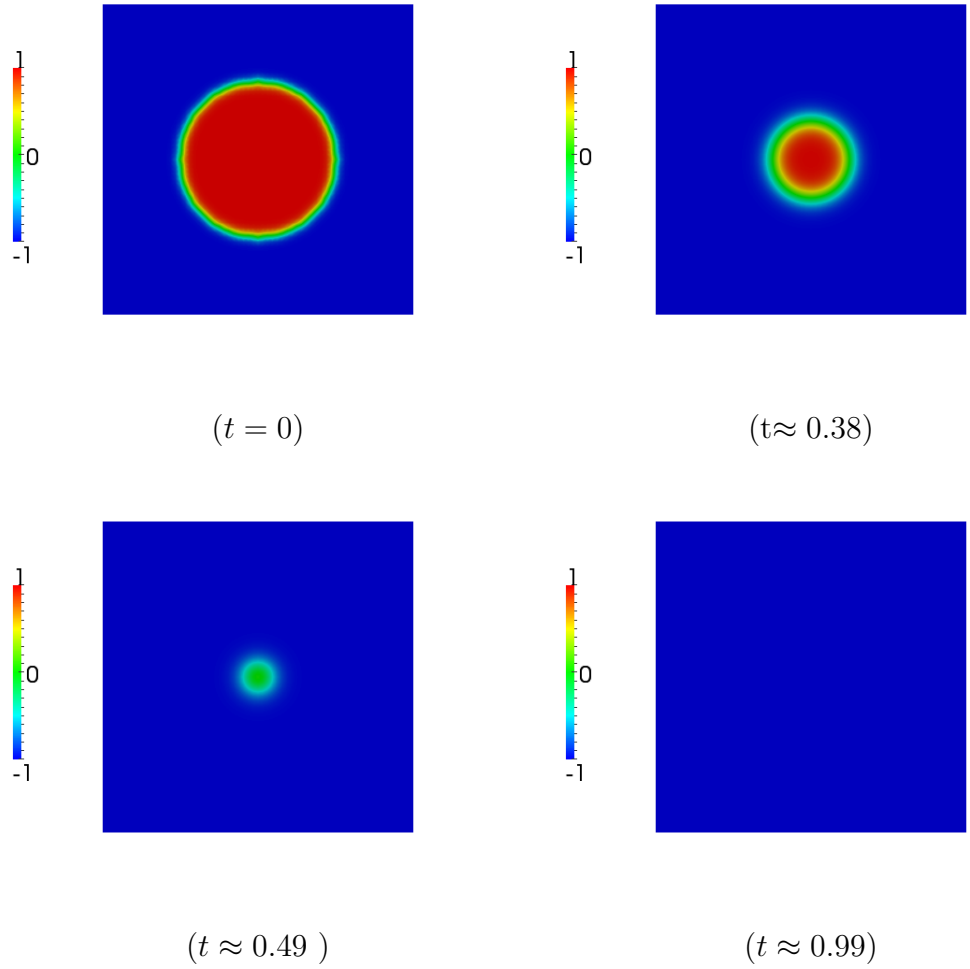


Figure 4.1: Evolution by mean curvature of a circle centred at the origin $(0, 0)$ with initial radius $R_0 = 1$ at different timesteps.

An alternative fundamental manner to validate the part of the proposed algorithm which is associated with the forward problem is to consider the "motion" of the unit circle to be governed by the equation (4.5) taking $\lambda_V = 0$ and

$$\eta(\boldsymbol{x}, t) = H(\boldsymbol{x}, t),$$

where $H(\boldsymbol{x}, t)$ is the mean curvature of the circle.

In this case it is expected that the circle will remain stationary at the same position, since $\boldsymbol{V}(\boldsymbol{x}, t) = 0$ (see Figures 4.3 and 4.4). Then the phase-field model

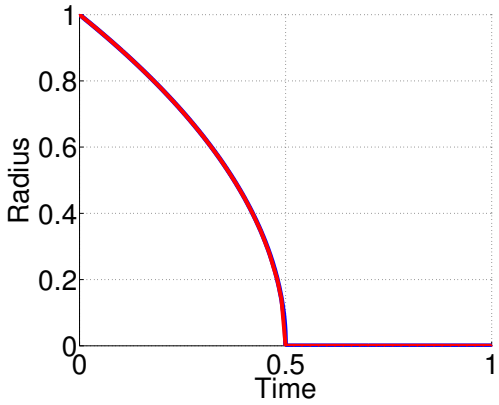


Figure 4.2: The radius $R(t) = \sqrt{R_0^2 - 2t}, \quad 0 \leq t \leq \frac{R_0^2}{2}$ of a unit circle that evolves by its mean curvature (blue line). The computed radius of the zero level-set of the solution of the Allen-Cahn equation (4.30) (red dotted line).

that approximates the equation of motion $\boldsymbol{V}(\boldsymbol{x}, t) = -H(\boldsymbol{x}, t) + \eta(\boldsymbol{x}, t)$ reads

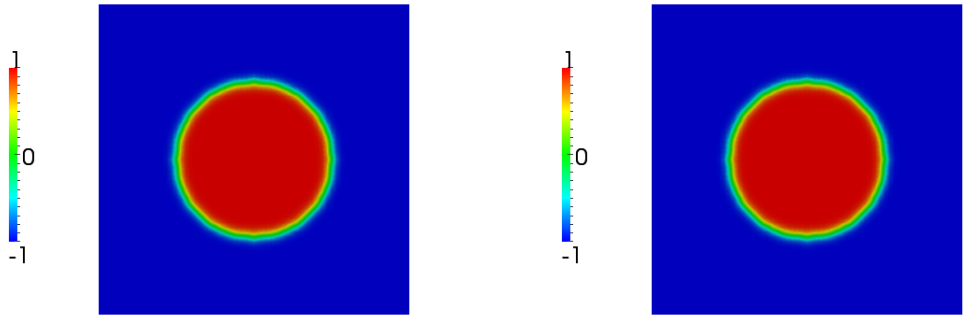
$$\begin{cases} \varepsilon \partial_t \varphi(\boldsymbol{x}, t) & = \varepsilon \Delta \varphi(\boldsymbol{x}, t) - \frac{1}{\varepsilon} G'(\varphi(\boldsymbol{x}, t)) - c_G \eta(\boldsymbol{x}, t) \text{ in } \Omega \times (0, T], \\ \nabla \varphi \cdot \boldsymbol{\nu}_\Omega & = 0 \text{ on } \partial\Omega \times (0, T], \\ \varphi(\boldsymbol{x}, 0) & = x_1^2 + x_2^2 - 1 \text{ in } \Omega. \end{cases} \tag{4.31}$$

where c_G is given by the formula (4.11). The rest of the parameter values used are presented in Table 4.1. Figure 4.3 shows that the initial circle remains stationary. This evolution is illustrated in Figure 4.4, in which the radius of the zero level-set of the computed solution remains unchanged.

A more complicated control problem for benchmarking is that of circle expansion. The initial phase-field function represents a circle centred at the origin $(0, 0)$ with radius $R_0 = 1$ while the desired circle has radius $R_{des} = 1.2$ with center at the origin $(0, 0)$. In the state equation (4.9) we set $\varepsilon = 0.1$, and we took a uniform time step $\tau = 1 \times 10^{-3}$ and end-time $T = 0.4$. The rest of the parameter values used are presented in Table 4.2.

We start with an abstract value for the control function $\eta(\boldsymbol{x}, t) = 0$ for our simulation. The top row in Figure 4.5 shows the computed phase-field representation of the circle's position at different times and the bottom row within the same figure depicts the values for the corresponding control function in the whole domain Ω.

Figure 4.6(a) shows snapshots of the zero level-sets of the solutions computed



$(t = 0)$ $(t = 1)$

Figure 4.3: The phase-field representation of the "motion" of the unit circle that evolves according to the evolution law $\boldsymbol{V}(\boldsymbol{x}, t) = -H(\boldsymbol{x}, t) + \eta(\boldsymbol{x}, t)$ and taking $\eta(\boldsymbol{x}, t) = H(\boldsymbol{x}, t)$.

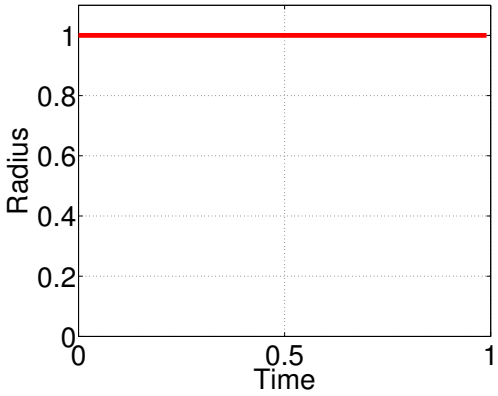


Figure 4.4: The computed radius of the zero contour of the phase-field representation. We observe that the radius of the level-set of the computed solution remains unchanged.

with the optimal control at different time steps. Figure 4.6(b) shows the plot of the objective functional against the number of iterations of the optimisation algorithm. The stopping criterion for this experiment was the change in the solution, through successive time steps, which was less than the predefined tol_{φ}.

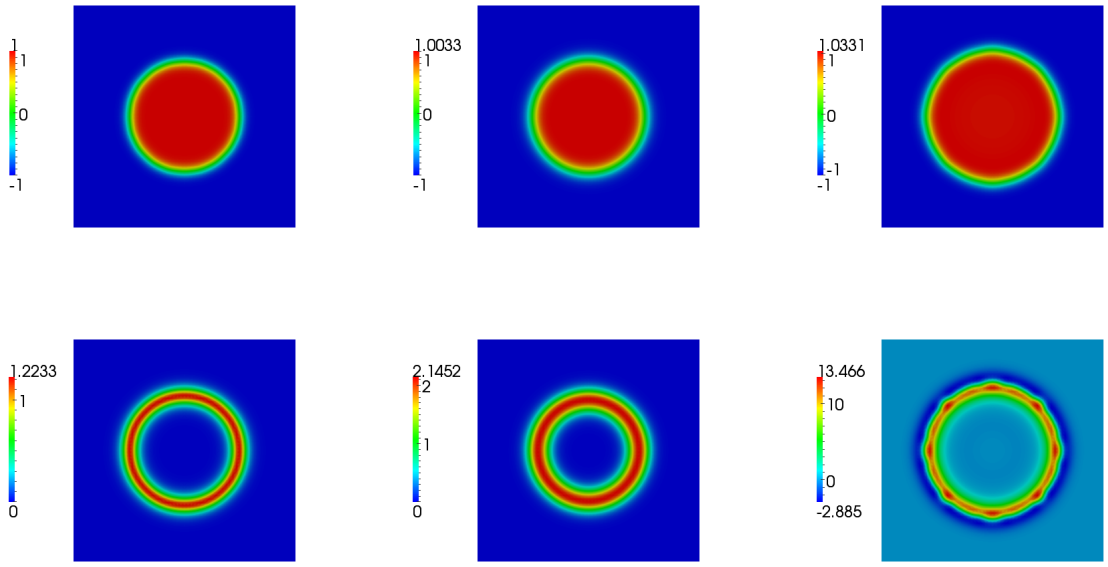
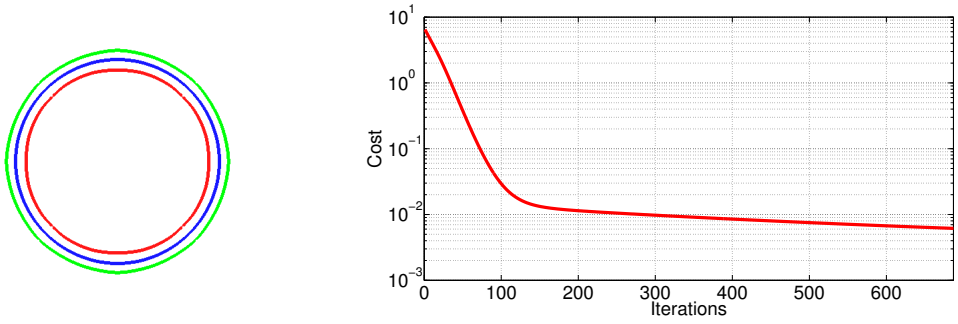


Figure 4.5: Top row: Computed phase-field representation at time $t = 0$, $t = \frac{T}{2}$ and $t = T$. Bottom row: Corresponding control function at time $t = 0$, $t = \frac{T}{2}$ and $t = T$.



(a) (b)

Figure 4.6: (a) Zero level-sets of the solutions computed with the optimal control for the example with the expanding circle after 0 (red), 340 (blue) and 400 (green) time steps. (b) The value of the cost functional versus the number of iterations for the example with the expanding circle.

4.5.2 Application to synthetic image data sets

We now present some numerical examples illustrating the application of the algorithm to synthetic image data sets. For all the simulations that we exhibit in this section, in the state equation (4.9) we set $\varepsilon = 0.1$, and we take the end-time

$T = 0.05$. The end time T corresponds to the nondimensionalised time between snapshots and could in principle be related to an acquisition time between images given real biological data. For each of the experiments we set the initial guess for the control to be constant in space and time (zero for the single cell case and one for the multi-cell examples). For the approximation of the forward and adjoint partial differential equations we use a triangulation with 8321 degrees of freedom and select a uniform timestep $\tau = 1 \times 10^{-4}$. The same numerical parameters for the optimisation algorithm were used for all the experiments and are given in Table 4.2.

α	θ	tol_J	tol_η	tol_φ	K_{max}	tol_λ
0.01	0.01	1×10^{-4}	1×10^{-4}	5×10^{-5}	3500	1×10^{-2}

Table 4.2: Parameter values used for the experiment with the expanding circle in §4.5.1 and the numerical simulations in §4.5.2.

Application to synthetic data

Here we apply the algorithm to a single synthetic cell data set taken from the *PhagoSight* website `http://www.phagosight.org/synData.php`. *PhagoSight* provides synthetic data sets representing neutrophils as observed in a Zebrafish embryo (Henry et al. [2013b]). The data for our computations consist of two-dimensional points on the synthetic cell membrane at a series of times. The initial and target curves we take as test data for the algorithm are shown in Figure 4.7(a). To apply the algorithm, based on diffuse interface representations, we define the domain $\Omega{:=}[0, 8] \times [0, 6]$ which is such that both the initial and target curves are contained in the domain.

We then define a diffuse interface approximation that describes the interior and the exterior region of the cell membrane following the procedure described in Croft et al. [2013]. Both are separated from each other by a diffuse interface, which denotes the cell membrane. We define the bulk domain $\Omega = \Omega_{ext} \cup \Gamma(t) \cup \Omega_{int}$ in which the

phase-field variable $\varphi(\boldsymbol{x}, t)$ is defined as

$$\varphi(\boldsymbol{x}, t) = \begin{cases} 1 & \text{if } d_{\Gamma(t)}(\boldsymbol{x}) > \varepsilon, \\ \sin\left(\frac{\pi d_{\Gamma(t)}(\boldsymbol{x})}{2\varepsilon}\right) & \text{if } \left|d_{\Gamma(t)}(\boldsymbol{x})\right| < \varepsilon, \\ -1 & \text{if } d_{\Gamma(t)}(\boldsymbol{x}) < -\varepsilon, \end{cases} \tag{4.32}$$

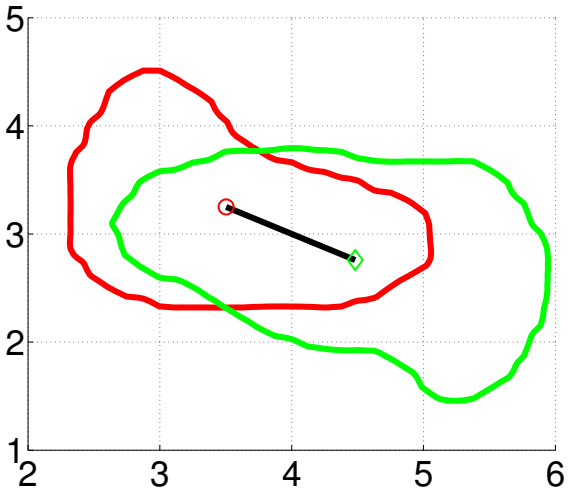
where ε denotes the interfacial thickness of the diffuse interface, and $d_{\Gamma(t)}(\boldsymbol{x})$ denotes the Euclidean signed-distance function between $\boldsymbol{x} \in \Omega$ and the closest point on $\Gamma(t)$ (Deckelnick et al. [2005]). The function $\varphi(\boldsymbol{x}, t)$ takes values $+1$ if the point is located in the interior of the cell and -1 if it belongs outside of the cell. The cell membrane $\Gamma(t)$ is implicitly defined as the zero level set of the $\varphi(\boldsymbol{x}, t)$.

Remark 20. *We note that other appropriate functions can be used for the definition of the phase-field variable $\varphi(\boldsymbol{x}, t)$, e.g. the cosine function.*

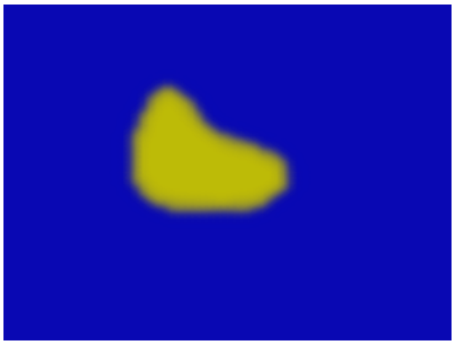
Figure 4.7 shows the diffuse interface representations of the initial (Figure 4.7(b)) and target data (Figure 4.7(c)), respectively.

In order to investigate the influence of the volume constraint on the computed cell morphologies we perform two experiments, in the first example we simply consider the forced Allen-Cahn model for the evolution with no volume constraint and in the second example we include the volume constraint as described in §4.2. The algorithm takes 1580 iterations to meet the stopping criteria with no volume constraint and 1740 iterations with the volume constraint, corresponding to CPU times of 27782 and 103024 seconds, respectively.

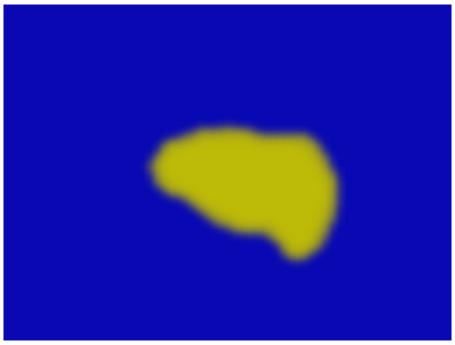
Figure 4.8 shows the plot of the objective functional against the number of iterations of the optimisation algorithm with and without the volume constraint. We observe an initial rapid decay in the objective functional for the case with the volume constraint whereas in the simulation with no volume constraint there is a gradually reduction until we approach the minimum. Figure 4.9 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint shaded by the values of the control. The background shading corresponds to the target data. In both cases the position of the zero level-set of the computed solution shows good agreement with the target data. Qualitatively, we observe cells with a clearly defined "front" and "rear", with the computed control



(a) Initial (red curve) and target (green curve) synthetic data. The cell centroids are shown together with the trajectory of the linear interpolant of the cell centroids (black line).



(b) Initial data (φ^0).

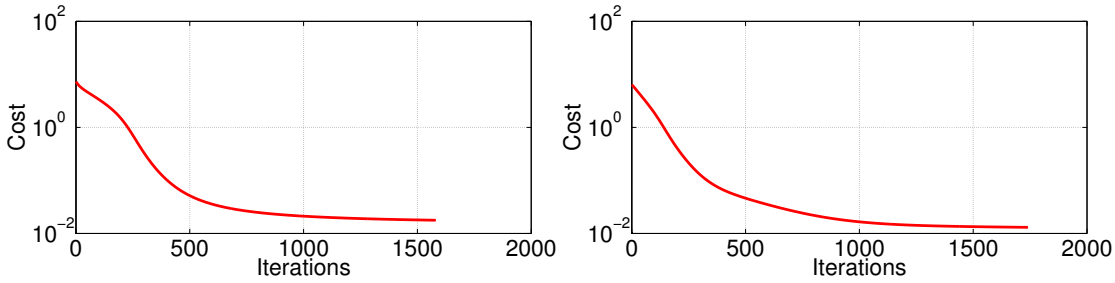


(c) Target data (φ_{obs}).

Figure 4.7: Initial and target data for the example with synthetic data from §4.5.2.

corresponding to protrusive forces at the front and contractive forces at the rear (Flaherty et al. [2007]; Xue et al. [2010]).

In Figure 4.10 we present the plot of the computed mass M_φ (see (4.12) for its definition) of the optimal solution, with the volume constraint. We observe that the computed mass is the linear interpolant of the data. Figure 4.11 shows the area enclosed by the zero level-set of the solution of the optimal control with and without the volume constraint together with the linear interpolant of the areas of

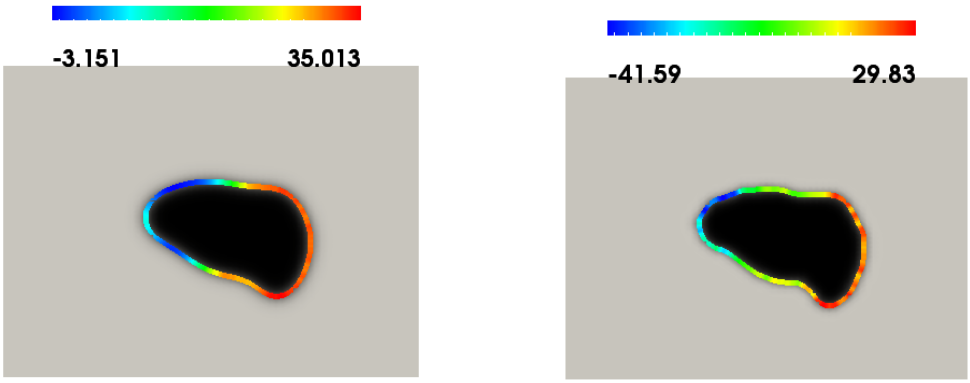


(a) Without the volume constraint (b) With the volume constraint

Figure 4.8: Plots of the cost functional versus the number of iterations for the experiments with synthetic data from §4.5.2, with and without the volume constraint. We observe an initial rapid decrease in the cost for the case with the volume constraint whereas in the simulation with no volume constraint is more gradually. Both followed by a much more steady decrease as we approach the minimum and this is as expected since the steepest descent algorithm is used for the update of the control.

the data. We see that without the volume constraint the area decreases linearly for long periods and at the final stages, the area increases exponentially to converge to the desired area. The algorithm is compensating for mismatching the area during the evolution process and only at the final stages does the algorithm converge to the desired shape. However, in the case with the volume constraint we observe that the area is close to the area generated by the linear interpolant of the data.

In terms of the computed cell morphologies, Figure 4.12 shows snapshots of the computed cell membranes (zero level-sets) for the two different cases. We clearly observe that the intermediate snapshot (blue curve) encloses a much smaller area if the volume constraint is not included in the algorithm. In Figure 4.13 we report on the trajectory of the centroid (center of mass) of the zero level-set of the computed solution of the optimal control, with and without the volume constraint. In addition, in Figure 4.14 we show the maximum and minimum values of the optimal control. We observe large increase in the maximum and minimum values of the control as we approach the final time.



(a) Without the volume constraint (b) With the volume constraint

Figure 4.9: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) with and without the volume constraint for the experiments with synthetic data from §4.5.2. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$.We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

4.5.3 The influence of the initial guess for the control

Here we apply the algorithm with the volume constraint on the simple example of a translated circle to illustrate the effect that the choice of the initial guess for the control η has on the solution of the problem. To apply our algorithm we define the domain Ω to be $[-3, 6] \times [-3, 3]$ with a triangulation of 8321 degrees of freedom. We selected a uniform timestep $\tau = 1 \times 10^{-3}$ and set the interfacial thickness $\varepsilon = 0.1$. We took the end-time $T = 0.8$. The remaining numerical parameters for the optimisation algorithm are as given in Table 4.1. The initial data was taken to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function taking the value 1 inside $B_1(0, 0)$ (a circle of radius 1 centred at the origin) and -1 in $\Omega \setminus B_1(0, 0)$. The target data was taken to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function taking the value 1 inside $B_1(3, 0)$ and -1 in $\Omega \setminus B_1(0, 0)$. Figure 4.15 shows the initial and target diffuse interface data. To illustrate the effect of the choice of initial guess on the algorithm, we consider

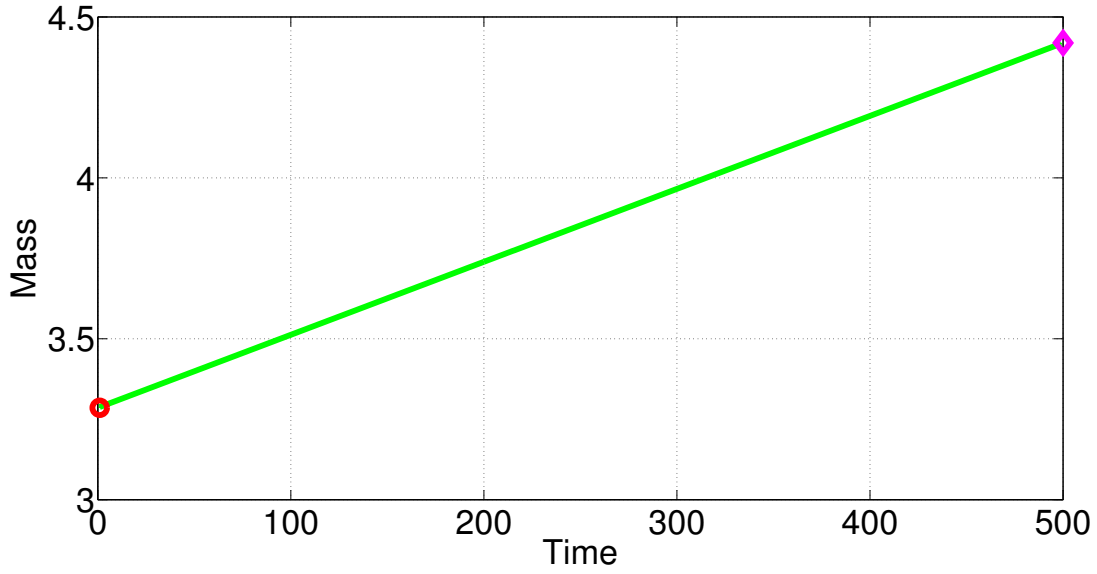


Figure 4.10: Plot of the computed mass M_φ for the experiment with synthetic data from §4.5.2 with the volume constraint. The circle and the diamond correspond to the mass of the initial and target diffuse interface data, respectively. We observe that the mass is the linear interpolant of the data. We refer to (4.12) for the definition of the M_φ.

two different values for the initial guess, firstly we set $\eta = 0$ and secondly we set $\eta = \boldsymbol{c}\cdot\nabla\varphi$, where $\boldsymbol{c} = (2.5, 0)$, i.e., in the latter case the initial guess depends on the solution to the Allen-Cahn equation. In both cases we used the algorithm with the volume constraints. With the zero initial guess the algorithm took 3262 iterations to meet the stopping criteria corresponding to a CPU time of 320433 seconds. With the second choice of initial guess the algorithm took 2056 iterations to meet the stopping criteria corresponding to a CPU time of 228173 seconds respectively.

Figure 4.16 shows the zero level-set of the computed solution using the optimal control at the final time. The curve corresponding to the zero level-set is shaded by the value of the control with the background shading corresponding to the target data. In both cases the position of the computed curve (zero level-set) with the optimal control shows good agreement with the target data. Figure 4.17 shows snapshots of the computed zero level-sets with the two different initial guesses. For the case with the initial value of $\eta = 0$, we observe in Figure 4.17(a) that the interface remains close to the initial position for most of the time of the simulation, and at the

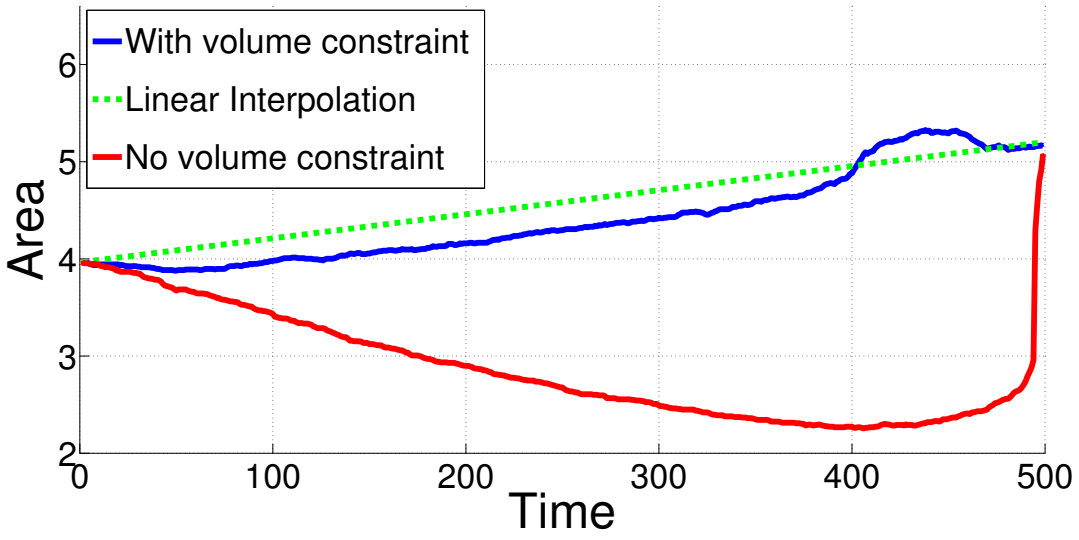
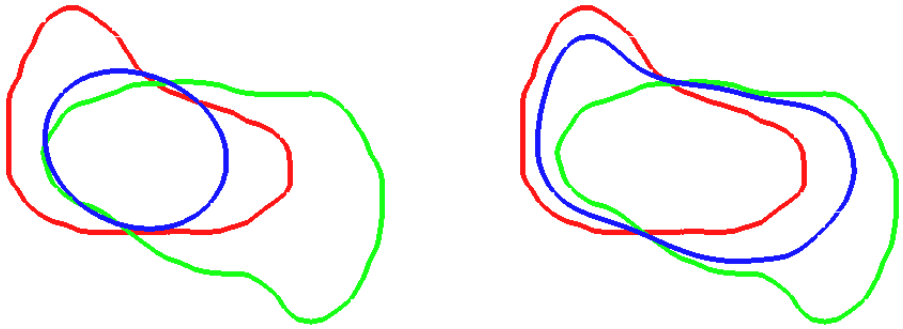


Figure 4.11: Plots of the area enclosed by the evolving cell for the experiments with synthetic data, with and without the volume constraint. The cell shrinks considerably during the evolution without the volume constraint leading to a mismatch in the target area. Only at the very last moment does the volume increases rapidly towards the target volume. The area enclosed from the simulation with the volume constraint is observed to be close to the linear interpolant of the data.

very last moment it shrinks to a point with a new phase nucleated at the position of the target data corresponding to a change in topology. With the second choice of initial guess ($\eta = \boldsymbol{c} \cdot \nabla\varphi$) we observe in Figure 4.17(b) that there is a gradual motion towards the target position with no changes in topology. Figure 4.18 shows the area enclosed by the zero level-set of the computed solution with the optimal control with the two different initial guesses together with the linear interpolant of the areas of the data. We observe a sharp increase in area towards the end of the time interval with the zero initial guess as the new phase is nucleated. With the second choice of initial guess, the area of the computed curve exhibits a good fit to the linear interpolant of the areas of the data.

Application to multi-cell image synthetic data sets

We now apply the algorithm to the case of multi-cell image data sets. As a proof-of-concept we consider the simplest possible scenario where we have an initial and



(a) Without the volume constraint (b) With the volume constraint

Figure 4.12: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments with synthetic data from §4.5.2, with and without the volume constraint after 0 (red), 350 (blue) and 500 (green) time steps. We observe that the volume enclosed by the blue curve is significantly smaller than the volumes enclosed by the red and green curves without the volume constraint whilst this is not observed if the volume constraint is included.

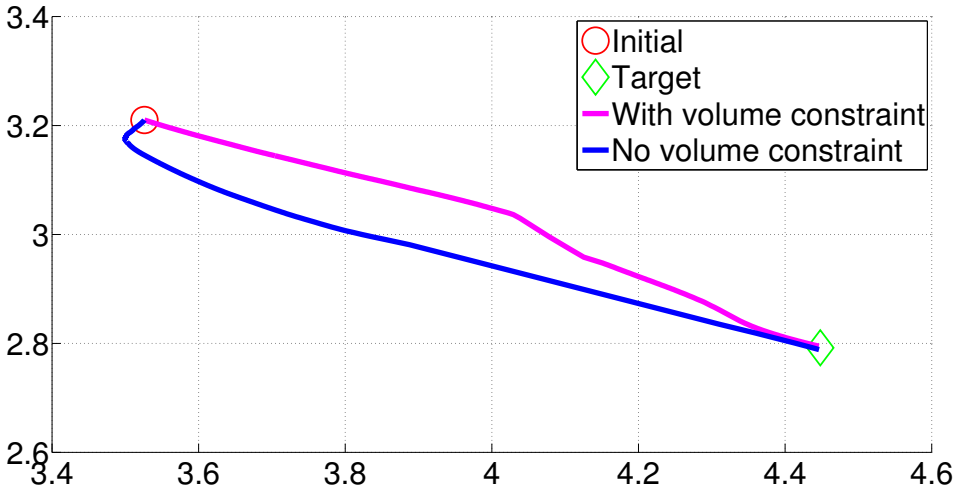
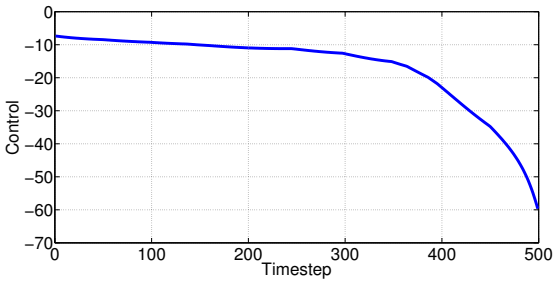
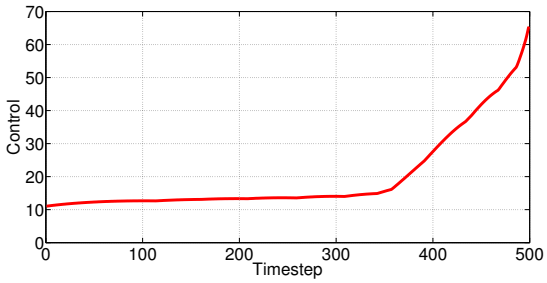


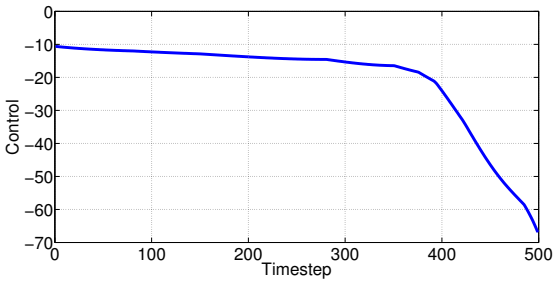
Figure 4.13: Trajectory of the centroid of the zero level-sets of the solution with the optimal control with and without the volume constraint for the experiments with synthetic data from §4.5.2.



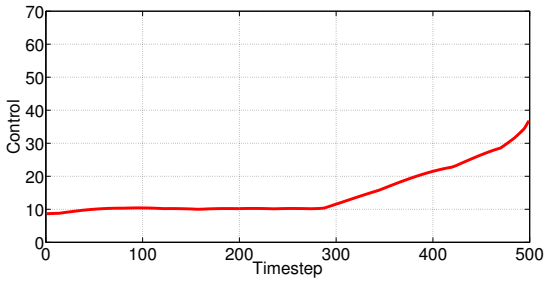
(a) Minimum value of the optimal control for the algorithm without volume constraints.



(b) Maximum value of the optimal control for the algorithm without volume constraints.

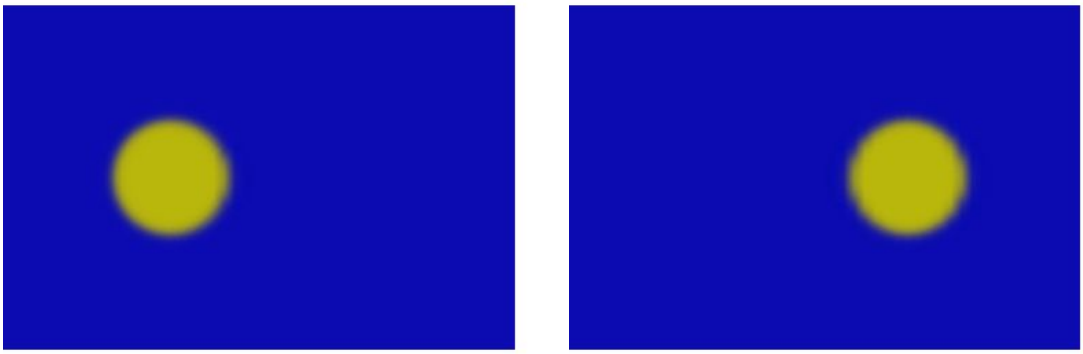


(c) Minimum value of the optimal control for the algorithm with volume constraints.



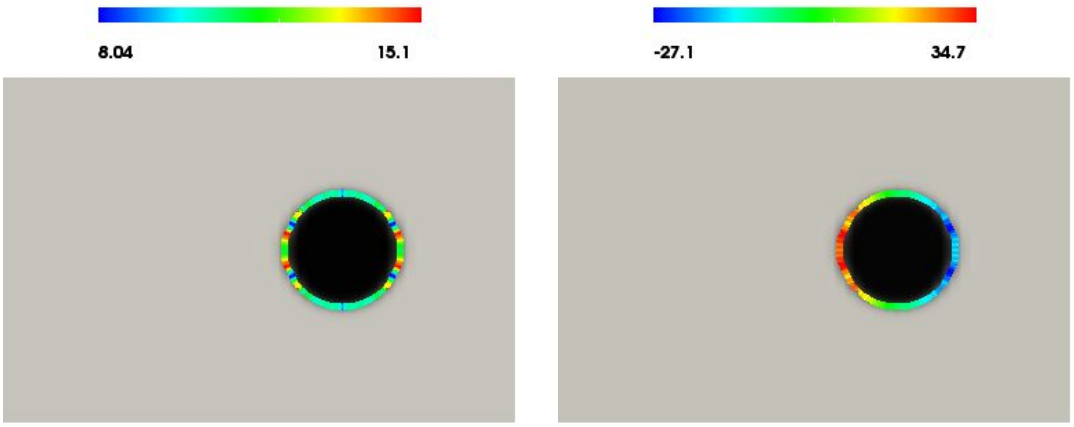
(d) Maximum value of the optimal control for the algorithm with volume constraints.

Figure 4.14: Minimum and maximum values of the control η with and without the volume constraint for the example with synthetic data from §4.5.2. We observe a large increase in the maximum and minimum values of the control, as we approach the final time of the simulations.



(a) Initial data (φ^0). (b) Target data (φ_{obs}).

Figure 4.15: Initial and target data for the examples of §4.5.3.

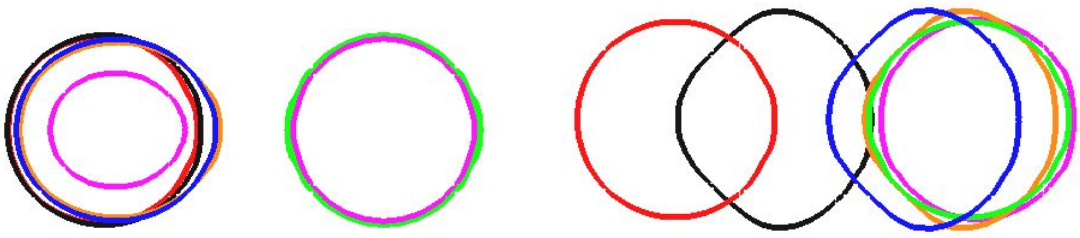


(a) With initial guess $\eta = 0$ (b) With initial guess $\eta = \boldsymbol{c} \cdot \nabla\varphi$

Figure 4.16: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments of §4.5.3. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

desired data set both consisting of two cells that are well separated.

For the first experiment we define the initial data and target data as follows. Defining the domain Ω to be $[-2, 8] \times [-2, 2]$ we construct the subdomains Ω_1,



(a) With initial guess $\eta = 0$ (b) With initial guess $\eta = \boldsymbol{c} \cdot \nabla\varphi$

Figure 4.17: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments of §4.5.3 at $t = 0$ (red), $t = 0.2$ (black), $t = 0.6$ (blue), $t = 0.7$ (orange), $t = 0.789$ (pink) and $t = 0.8$ (green). We observe the nucleation of a phase and a change in topology with the zero initial guess whilst there are no evident changes in topology and the zero level-set maintains a fixed topology in the case of the nonzero initial guess.

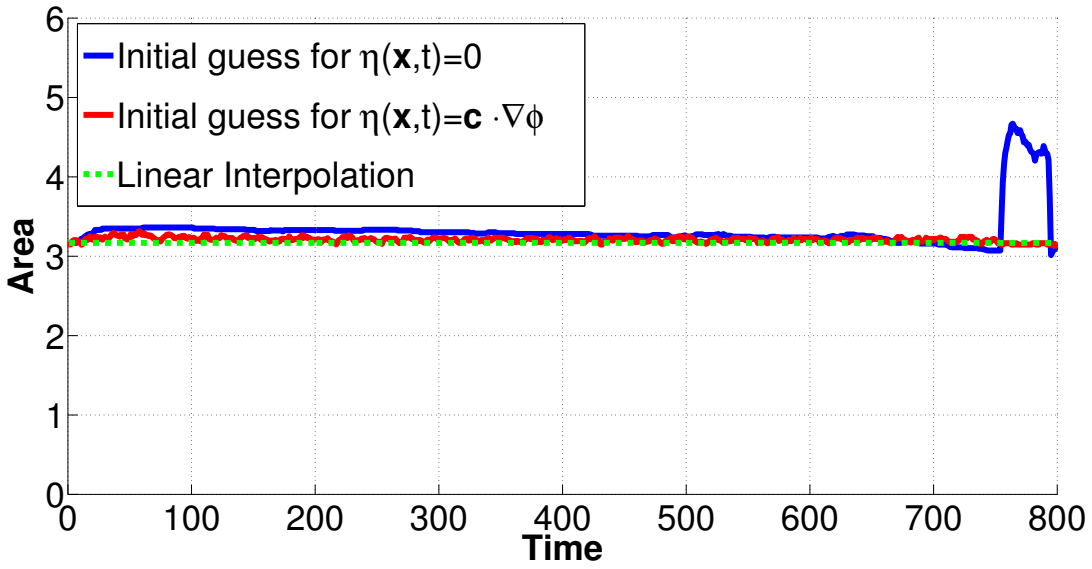


Figure 4.18: Area enclosed by the curve for the experiments of §4.5.3. A good fit to the linear interpolant of the areas is only observed with the nonzero initial guess. We observe a rapid increase in the area near the end time for the zero initial guess, this corresponds to the time at which a new phase is nucleated, c.f., Figure 4.17(a).

Ω_2, Ω_3 and Ω_4 to be the simply connected bounded domains with boundary curves $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 defined by (the curves Γ_1, Γ_2 and Γ_3 and Γ_4 are the zero level-sets of the diffuse interfaces shown in Figure 4.19(a) and 4.19(b), respectively).

$$\Gamma_1 := \left\{x \in \Omega \,|\, x_1^2 + x_2^2 - 0.8^2 + 0.1\sin(4x_1) + 0.1\sin(3x_2) = 0\right\},$$
$$\Gamma_2 := \left\{x \in \Omega \,|\, \left(\frac{x_1}{2} - 2\right)^2 + (x_2 - 0.6)^2 - 0.7^2 + 0.1\sin\left(\frac{5x_1}{2}\right) + 0.3\sin(2x_2) = 0\right\},$$
$$\Gamma_3 := \left\{x \in \Omega \,|\, (x_1 - 0.4)^2 + (x_2 - 0.5)^2 - 0.8^2 + 0.1\sin(6x_1) + 0.1\sin(7x_2) = 0\right\},$$
$$\Gamma_4 := \left\{x \in \Omega \,|\, \left(\frac{x_1}{2} - 2.5\right)^2 + (x_2 - 1)^2 - 0.7^2 + 0.1\sin\left(\frac{7x_1}{2}\right) + 0.1\sin(1.5x_2) = 0\right\}.$$

We then set the initial and target data to be a smoothed (by running 5 steps of the Allen-Cahn solver) version of the function

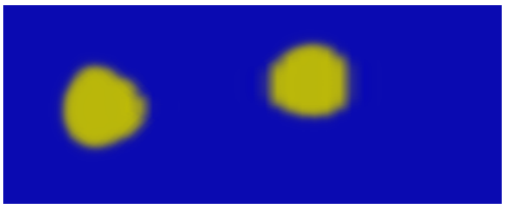
$$\varphi^0 = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_1 \cup \Omega_2, \\ -1 \text{ for } \boldsymbol{x} \in \Omega \setminus (\Omega_1 \cup \Omega_2), \end{cases} \quad \text{and} \quad \varphi_{obs} = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_3 \cup \Omega_4, \\ -1 \text{ for } \boldsymbol{x} \in \Omega \setminus (\Omega_3 \cup \Omega_4). \end{cases}$$

Figure 4.19 shows the initial and target diffuse interface data.

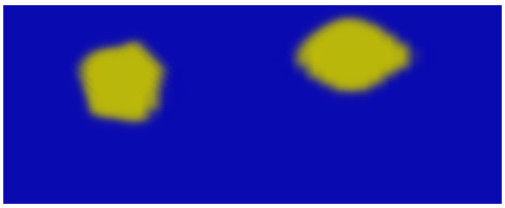
As previously, we compare the results of the algorithm with and without the volume constraint. For this experiment, the algorithm takes 1149 iterations to meet the stopping criteria with no volume constraint and 3013 iterations with the volume constraint, corresponding to CPU times of 22132 and 207663 seconds, respectively.

Figure 4.20 shows the plot of the cost functional against the number of iterations of the optimisation algorithm with and without the volume constraint. Figure 4.21 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint shaded by the values of the control with the background shading corresponding to the target data. The results are similar to the single cell simulations in the previous example, with an initial rapid decrease in the cost followed by a subsequent gradual decrease. The cells (zero level-sets) computed with the optimal control show good agreement with the target data for both versions of the algorithm and for both cells at the final time as expected, but not for the intermediate positions.

Figure 4.22 shows the area of the domain in which the computed solution is positive with and without the volume constraint together for the case of the linear interpolant of the areas of the data. We observe analogous behaviour to the single



(a) Initial data (φ^0).



(b) Target data (φ_{obs}).

Figure 4.19: Initial and target data for the examples with the multi-cell image data sets of §4.5.3.

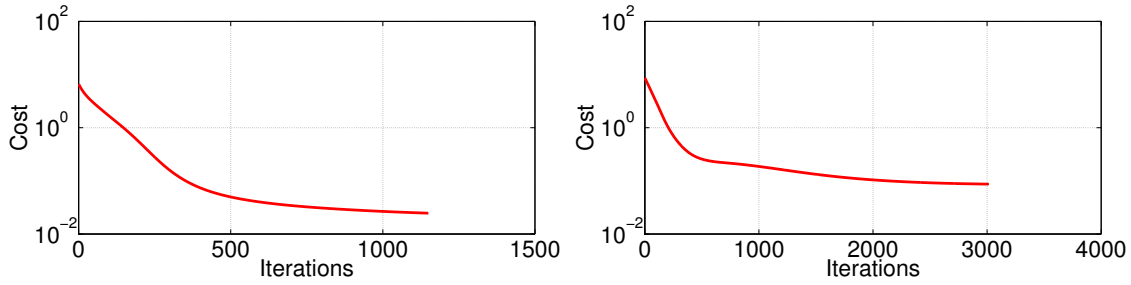
cell. In terms of the computed cell morphologies, Figure 4.23 shows snapshots of the computed zero level-sets for the two different versions of the algorithm.

We see that in this multi-cell setting the algorithm has implicitly solved the matching problem by generating two disjoint cells. We observe that the loss of volume in the case of no volume constraint corresponds to one of the cells in the intermediate snapshot (blue curve) enclosing a much smaller area.

An example with topological change

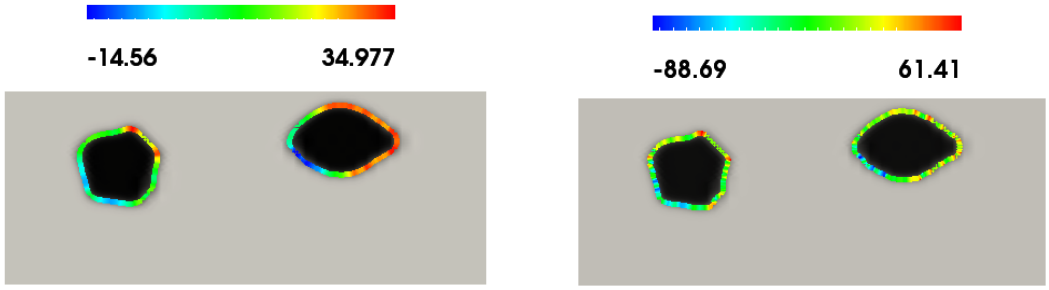
It must be noted that, in general our algorithm may generate cells whose topology is not fixed. In the multi-cell setting it is very easy to generate examples for which we observe changes in topology in the cell membrane as illustrated by the next experiment.

Defining the domain Ω to be $[-2, 6.3] \times [-2.5, 2.5]$ we construct the subdomains



(a) Without the volume constraint (b) With the volume constraint

Figure 4.20: Cost functional versus the number of iterations for the examples with the multi-cell image data sets of §4.5.3.



(a) Without the volume constraint (b) With the volume constraint

Figure 4.21: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) with and without the volume constraint for the experiments with the multi-cell image data sets of §4.5.3. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. For both cases, we see good agreement between the zero level-set of the data computed with the optimal control and the target data.

Ω_1, Ω_2, Ω_3 and Ω_4 to be the simply connected bounded domains with boundary curves $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 defined by (see Figure 4.24)

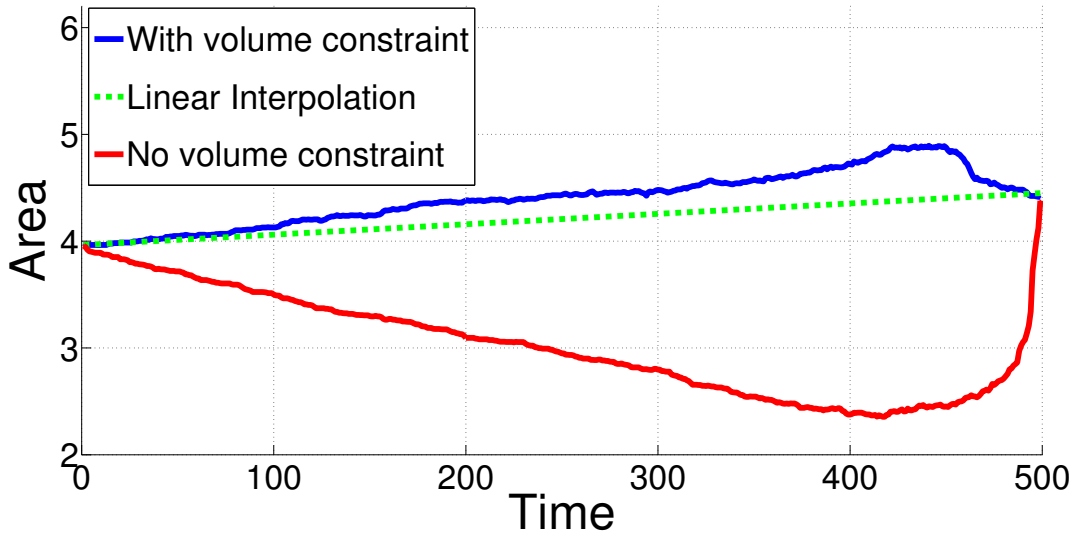


Figure 4.22: Plots of the area enclosed by the cell for the experiments for the case of the multi-cell image data sets of §4.5.3, with and without the volume constraint. As with the single cell data, the area (now the sum of the areas of the two cells) shrinks considerably during the evolution without the volume constraint whilst a better fit to the linear interpolant of the area enclosed by the data is observed with the volume constraint.

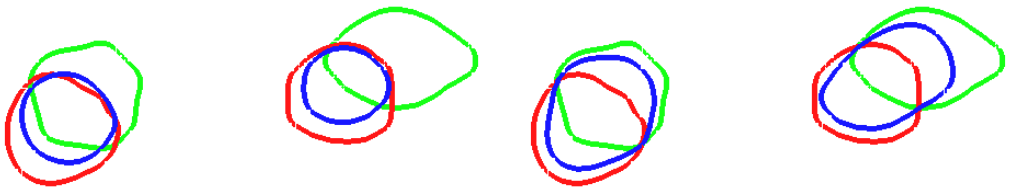
$$\Gamma_1 := \left\{ x \in \Omega \,|\, x_1^2 + x_2^2 - 0.9^2 + 0.1\sin(4.5x_1) + 0.11\sin(3x_2)) = 0 \right\},$$
$$\Gamma_2 := \left\{ x \in \Omega \,|\, (x_1 - 5)^2 + x_2^2 - 0.7^2 + 0.1\sin\left(\frac{5x_1}{2}\right) + 0.3\sin(2x_2) = 0 \right\},$$
$$\Gamma_3 := \left\{ x \in \Omega \,|\, (x_1 - 0.35)^2 + (x_2 - 0.7)^2 - 0.8^2 + 0.1\sin(6x_1) + 0.1\sin(7x_2) = 0 \right\},$$
$$\Gamma_4 := \left\{ x \in \Omega \,|\, (x_1 - 0.3)^2 + (x_2 - 1.1)^2 - 0.7^2 - 0.1\sin\left(\frac{7x_1}{2}\right) + 0.1\sin(1.5x_2) = 0 \right\}.$$

We then set the initial and target data to be a smoothed (by running 5 steps of the Allen-Cahn solver) version of the function

$$\varphi^0 = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_1 \cup \Omega_2, \\ -1 \text{ for } \boldsymbol{x} \in \Omega \setminus (\Omega_1 \cup \Omega_2), \end{cases} \quad \text{and} \quad \varphi_{obs} = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_3 \cup \Omega_4, \\ -1 \text{ for } \boldsymbol{x} \in \Omega \setminus (\Omega_3 \cup \Omega_4). \end{cases}$$

Figure 4.24 shows the initial and target diffuse interface data.

Similar to previous experiments, we compare the results of the algorithm with and without the volume constraint. For this experiment, the algorithm takes 1776



(a) Without the volume constraint (b) With the volume constraint

Figure 4.23: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the multi-cell image data sets of §4.5.3, with and without the volume constraint after 0 (red), 350 (blue) and 500 (green) time steps. The volume enclosed by both cells shrinks during the evolution without the volume constraint whilst this is not observed if the volume constraint is included. For both cases, with and without the volume constraint, the implicit solution of the matching problem in this case generates two disjoint cells which do not change in topology.

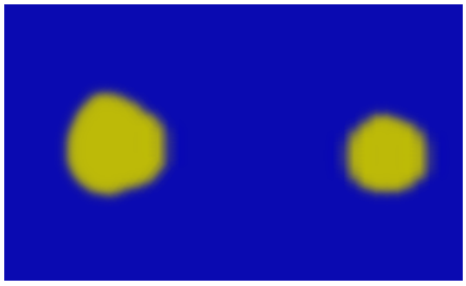
iterations to meet the stopping criteria with no volume constraint and 2813 iterations with the volume constraint, corresponding to CPU times of 33290 and 193476 seconds, respectively.

Figure 4.25 shows the plot of the objective functional against the number of iterations of the optimisation algorithm with and without the volume constraint. Figure 4.26 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint shaded by the values of the control where the background shading corresponding to the target data. The results are similar to the previous simulations with an initial rapid decrease in the cost function followed by a subsequent gradual decrease and good agreement with the target data for both versions of the algorithm and for both cells. For each of the versions of the algorithm, both of the computed cells posses a clearly defined "front" and "rear" of the evolving cell.

Figure 4.27 shows the area of the domain in which the computed solution is positive with and without the volume constraint together with the linear interpolant of the areas of the data. Figure 4.28 shows snapshots of the computed zero level-sets

for the two different versions of the algorithm. Unlike in the previous examples we see that for this particular choice of initial and target data, the algorithm yields cells which change in topology with one of the curves shrinking until it disappears whilst the other curve splits into two disjoint curves. Thus our algorithm generates trajectories corresponding to the annihilation (via shrinking) of one cell whilst the other cell splits to form the two cells observed in the image data set.

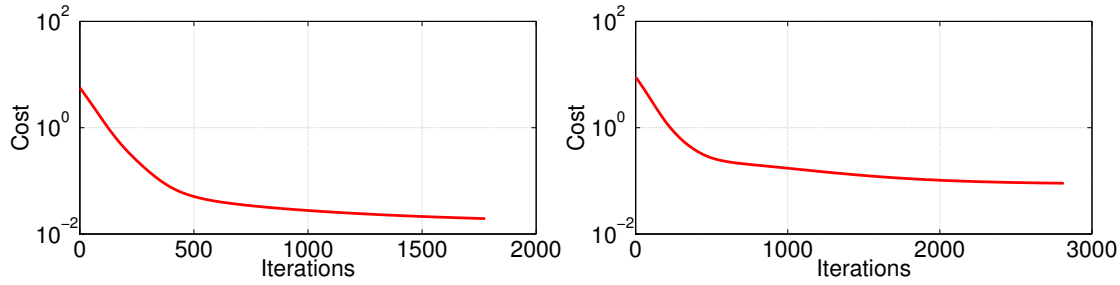
Remark 21. *Biologically this process described above can be explained by the phenomenon of mitosis and apoptosis. The former refers to the cell's cycle process in which a cell divides into two or more cells. The latter describes the procedure of programmed cell death and one of the changes involved include cell shrinkage among others (Alberts et al. [1994]).*



(a) Initial data (φ^0).

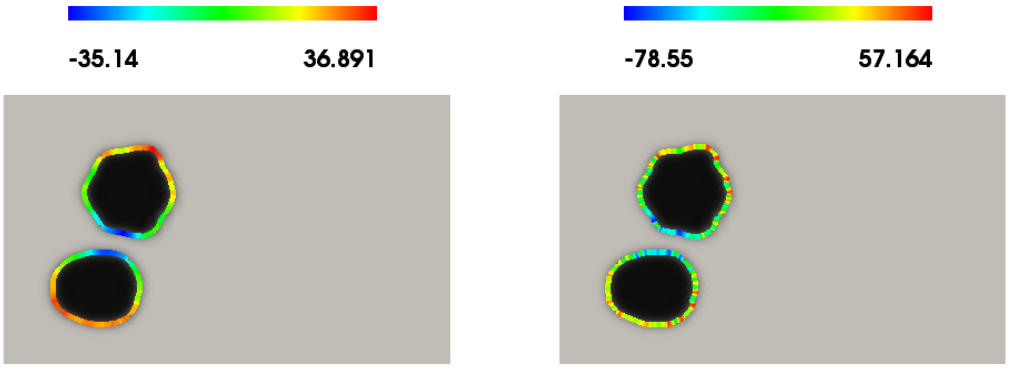
(b) Target data (φ_{obs}).

Figure 4.24: Initial and target data for the examples with topological change of §4.5.3.



(a) Without the volume constraint (b) With the volume constraint

Figure 4.25: The value of the cost functional versus the number of iterations for the examples of §4.5.3 with and without the volume constraint. We observe a rapid decrease in the cost initially followed by a much more gradual decrease as we approach the minimum, this is as expected since the steepest descent algorithm is used for the update of the control.



(a) Without the volume constraint (b) With the volume constraint

Figure 4.26: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) with and without the volume constraint for the experiments with topological change of §4.5.3. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

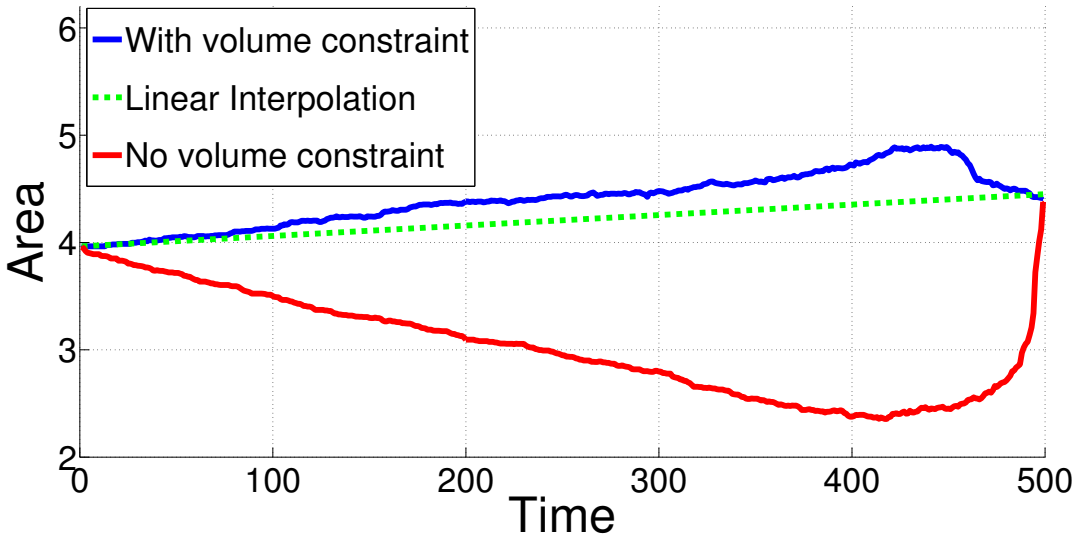
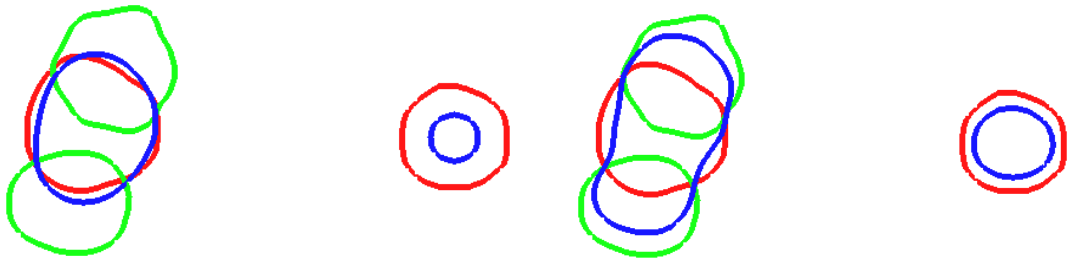


Figure 4.27: Area enclosed by the cell for the experiments with topological change of §4.5.3 with and without the volume constraint. As with the single cell data, the area (now the sum of the areas of the two cells) shrinks considerably during the evolution without the volume constraint. However, the area enclosed by the cell with the volume constraint does exhibit a better fit to the linear interpolant of the areas of the data.

4.5.4 Comments and limitations of the cell tracking algorithm

Like other algorithms (Henry et al. [2013b]), this proposed algorithm for cell tracking with the volume constraint has limitations. In Figures 4.11, 4.22 and 4.27 we observe that the area enclosed by the evolving cells, when it approaches the end time, increases and does not exhibit a good fit to the linear interpolant of the areas of the data. Looking at the phase field representation of the computed solution (Figure 4.29), as shown from the experiment with the synthetic data from §4.5.2, we see that the two-phase interface is not "well" preserved after 420 time steps. The values of the phase-field variable $\varphi(\boldsymbol{x}, t)$ are no longer around -1 and $+1$ but fluctuates between -0.994 and 1.125. A potential reason for this may be the high values of the control function $\eta(\boldsymbol{x}, t)$.

Nevertheless, it is expected to solve this problem by reducing the interfacial thickness ε and by increasing the number of degrees of freedom (DOFs). However, this becomes computationally prohibitive regarding the memory requirements of the



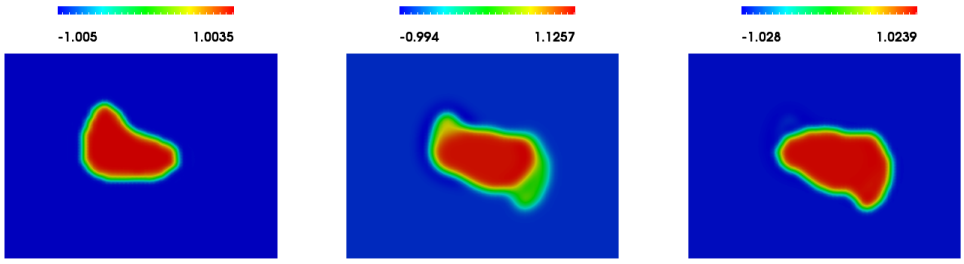
(a) Without the volume constraint (b) With the volume constraint

Figure 4.28: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments with topological change as described in §4.5.3 with and without the volume constraint after 0 (red), 350 (blue) and 500 (green) time steps. We observe that both with and without the volume constraint, the implicit solution of the matching problem in our algorithm leads to the annihilation of one cell (as it shrinks to a point) while the other cell splits with the zero level-set changing in topology from a single closed curve to two disjoint closed curves.

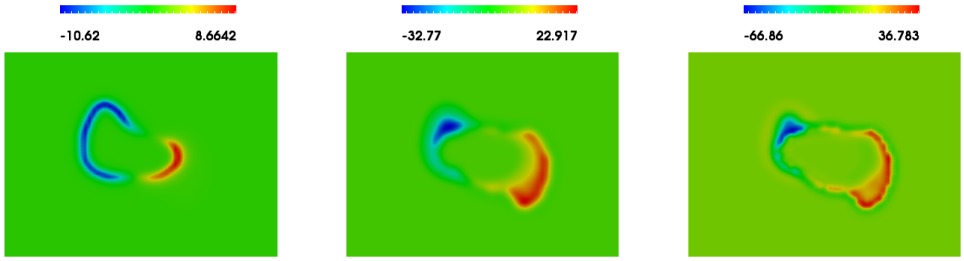
algorithm. A future area of research that could act as an alternative remedy for this issue would be to impose constraints on the control function $\eta(\boldsymbol{x}, t)$.

An additional limitation for the proposed algorithm for cell tracking with the volume constraint is the time that is required for a simulation. The CPU times for each of the experiments above is on the order of hours. For all the experiments the number of iterations required before the stopping criteria is met are similar, however this leads to simulations with the volume constraint taking longer time (in terms of CPU time) as those without the volume constraint. This is due to the iterative nature of the algorithm used to compute the Lagrange multiplier c.f. §4.3.1 which necessitates multiple solves per time step. We note that the CPU times of the algorithms may be too large for many applications.

Finally, we mention that the current solution procedure based on grids where a constant mesh connectivity throughout the simulations has been used and serial solution of the forward and adjoint problems may be improved by combining adapt-



(a) Phase-field representation



(b) Control function

Figure 4.29: (a): Computed phase-field representation after 0, 420 and 500 time steps, for the example with the synthetic data from §4.5.2 with the volume constraint. (b): Corresponding control function after 0, 420 and 500 time steps. We observe that the two-phase interface is not well preserved when we approach the end time (after 420 iterations) and the phase-field variable $\varphi(\boldsymbol{x}, t)$ takes values from -0.994 to 1.125. A possible reason for this may be the big values of the control function.

ive finite element grids with a parallel solver for the forward and adjoint problems. This will make our computations much quicker.

In addition, for this study we employ the steepest descend method to update (4.29) the control variable $\eta(\boldsymbol{x}, t)$. One feature of this method is that it is easy to implement. It must be noted that this method works well in the initial iterations, but tends to be very slow until it converges to the optimal solution (Wang [2008]).

4.6 Conclusion

In this chapter we presented a first step towards the development of cell tracking algorithms based on physical models for cell migration. The presented algorithm seeks

to track whole cell morphologies and is applicable to single cell or multi-cell image data sets. Our approach may be regarded as a model fitting procedure in which a physically derived model for the evolution of the cell or cells is fitted to experimental image data sets. The algorithm is based on the theory of optimal control of partial differential equations and full details of the derivation and implementation of the algorithm are given. We also present a number of numerical experiments illustrating the validation and the performance of the algorithm with synthetic representative single cell and multi-cell image data sets.

The key novelty of our approach is that the model for the evolution of the cell (or cells), which drives the tracking procedure, is based on a relevant simplification of existing physically derived models for cell motility that reproduce many experimentally observed aspects of cell migration (e.g. Neilson et al. [2010]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]). Thus this study is an important step towards the development of cell tracking algorithms in which the recovered trajectories are physically meaningful.

One significant advantage of this approach for cell tracking is that the physics of the model driving the evolution of the cell is reflected in the recovered dynamic data. Thus it is possible to encode physical features of cell migration into the tracking procedure. We illustrate this fact by including volume constraints in the model. Comparing the results of the tracking algorithm with and without volume constraints, we observe, that in a number of simulations neglecting volume constraints leads to physically unrealistic cell morphologies with a significant reduction of the cell volume in the recovered morphologies whilst this undesirable effect is no longer evident if volume constraint is included.

We worked with diffuse interface representations of the cell membrane to make use of the mature theory for the optimal control of semilinear partial differential equations. One attractive aspect of this approach is that, as we do not require sharp interface representations of the cell membrane, it may be possible therefore to work directly with the raw experimental image data set without any need for segmentation.

Like other algorithms, the proposed cell tracking algorithm has limitations. As we observed in Figure 4.29, the two-phase interface is not “well” preserved when we

approach the end time and we presume that a potential reason for this issue may be the high values of the control function. Thus, a future step for the improvement of this algorithm is to impose constraints on the control function $\eta(\boldsymbol{x}, t)$. In addition, the diffuse interface or phase-field framework that we employ does make the algorithm computationally intensive as evidenced by the relatively large CPU times for our experiments and a key area for future work is to investigate improvements in the computational efficiency of the algorithm. This need is especially evident if one wishes to track cells in three dimensions. Although our theoretical framework applies equally to this setting the computational cost becomes prohibitive. Computational aspects under investigation include

- Spatial and temporal mesh adaptivity which is challenging in this setting as the solution of the state equation enters the adjoint equation (Haußer et al. [2010]; Haußer et al. [2012]).
- Alternative update schemes for the control (4.29) to the simple yet robust gradient based update considered in this chapter. We can consider the second order optimisation Newton's method (Snyman [2005]). Second order methods often converge much more quickly, but they can be challenging in their implementation under the current framework.
- Parallelisation and the development of fast solvers for the solution of the state and adjoint equations.

Our initial numerical investigations suggest that with a combination of the techniques outlined above it is possible to efficiently track three-dimensional cell migration.

As mentioned previously one interpretation of the forcing η^* is that it accounts for both protrusive forces generated by polymerisation of actin at the leading edge of the cell together with contractile forces generated by the action of myosin motors at the cell rear. Thus, a potential avenue for assessing the plausibility of the cell tracks computed with our algorithm would be to compare the computed η^* with experimental imaging data on the location of polymerised actin and myosin-II on the cell membrane with the expectation being that regions in which the computed forcing η^* is positive would correspond to regions rich in polymerised actin and

regions in which the computed forcing η^* is negative would correspond to regions rich in myosin-II.

There are also many extensions of our approach which are likely to prove useful in applications. Our algorithm could equally be applied to the identification of (possibly time-dependent) parameters in models for cell migration (e.g., a spatially constant forcing or material parameters such as surface tension or bending rigidity), however, for these cases it is likely that the sharp interface approach proposed in Croft et al. [2013] will be more efficient. As observed in some of the experiments we report on, the framework we employ allows changes in topology of the cells. Whilst this may be desirable for some applications, e.g., tracking cells beyond cell division or cell fusion, in many biological experiments the topology of the cells does not change.

The model we propose for the evolution in this chapter is a simplification of more general physically relevant models in which bulk or surface partial differential equations for the biochemistry are coupled to a geometric evolution law for the motion. An important area for future work is the extension of the framework to this more general setting. We note that the phase-field approach we employ makes it computationally straightforward to couple the geometric evolution law for the motion to bulk partial differential equations (posed either within the cell or in the extra-cellular matrix) (Ziebert et al. [2011]; Shao et al. [2010]; Shao et al. [2012]).

Finally, investigating the performance of the proposed algorithm with real biological data for different cell types and in different environments is an important and worthwhile task. In the next chapter we present the results of the algorithm applied to the tracking of *in vivo* neutrophil migration as observed in the zebrafish *Danio rerio larvae*.

Chapter 5

An application of cell tracking algorithm in neutrophil cell migration

5.1 Introduction

Biological processes constitute the functional foundation for living organisms. Inflammation is such a process with which multicellular organisms can defend themselves against other competitive microorganisms or tissue injury (Henry et al. [2013b]). The inflammatory response initiates when a tissue injury is caused by bacteria, trauma, toxins or any other cause (Andreoli et al. [2010]). Chemicals are released attracting leukocytes such as monocytes, lymphocytes or neutrophils. The latter represent the most common immune cells and together with macrophages are the key elements for a successful immune response (Elks et al. [2011]). Our daily defence against pathogens depends on their action and their quick migrating response (Reyes-Aldasoro et al. [2009]). In addition, neutrophils are produced to kill bacteria in cases where macrophages fail to control the initial infection (Elks et al. [2011]).

Their motility is of vital importance and results from a coordination of protrusions (pseudopods) and retractions, leading to a deformation of their shape (Yap and Kamm [2005]; Flaherty et al. [2007]). The production of these pseudopods seems to be affected by the presence of chemical gradients and is subject to complex molecular interactions. For all these reasons, it becomes obvious why neutrophils play

a central role by providing a rapid immune response and therefore it is necessary to understand their behaviour.

The most common transgenic animal model that has been used in recent years for the above purpose is the zebrafish *Danio rerio larvae*. This model presents numerous advantages for their study (Renshaw et al. [2006]; Reyes-Aldasoro et al. [2009]; Elks et al. [2011]; Najia et al. [2011]; Holmes et al. [2012]; Kadirkamanathan et al. [2012]; Henry et al. [2013b]). Their immune system includes similar cell types close to those found in humans (Renshaw et al. [2006]; Henry et al. [2013a]). One of the main advantages of the zebrafish larvae is their transparency, facilitating the visualisation of a wide spectrum of processes with the aid of the differential interference contrast (DIC) microscopy (Elks et al. [2011]). This attribute combined with its transgenic nature, make it possible to be labelled with green fluorescent markers (Tg(mpx:eGFP)i114), and the feasible genetic manipulations classify zebrafish *Danio rerio larvae* as a powerful model system (Kadirkamanathan et al. [2012]).

Analytically, zebrafish has the ability to provide *in vivo* imaging for migrating cells, such as neutrophils. The use of differential interference contrast microscopy allows easy visualisation of the cell targets, resulting in quick experimental data. The induction of inflammation starts with the tail transection of zebrafish *Danio rerio larvae*. This action initiates the migration of neutrophils towards the injury site often modulated by the introduction of the chemoattractants. On contact with these or with pathogens neutrophils become "activated". It has been suggested that the deformation of their shape is considered as an early sign of this activation, though this relationship between these two features remains to be determined (Reyes-Aldasoro et al. [2009]). As a result, the importance of understanding neutrophil movement and behaviour for their defence against disease, as well as for the possible harmful outcome in cases of improper activation, it becomes obvious (Edwards [2005]; Reyes-Aldasoro et al. [2009]).

In Reyes-Aldasoro et al. [2009], studies have been attempted to track and perform a morphological analysis of the deformation of the three-dimensional neutrophil's shape during inflammatory process *in vivo*. A series of measurements have been proposed in order to relate the activation of the neutrophils and their shape modification. When neutrophils are activated and move towards the site of the injury

it has been observed that their volume increases. This has been demonstrated also in Henry et al. [2013b]. In addition, the volume has been noted to be correlated with the sphericity of neutrophils, which measures how compact or close they are to a perfect sphere. Later, Kadirkamanathan et al. [2012] presented a computational framework for the estimation and visualisation of the chemoattractant field using imaging data from tracked neutrophils. Their method is based on the assumption that there is a relationship between neutrophil's centroid velocity and the chemoattractant gradient.

In this chapter we will present an application of the optimal control algorithm (which was demonstrated in Chapter 4) to experimental data on the migration of neutrophils as observed in zebrafish *Danio rerio larvae*. In order to contrast this methodology, we compare the numerical results with those obtained in Chapter 3, where the cubic interpolation method was employed using cubic interpolation. Finally, a number of meaningful biological measures are calculated and compared, from both tracking methods. These measurements have been selected to demonstrate the potential of a morphological of neutrophils. We conclude this chapter by discussing the numerical results and suggesting possible research work related to the neutrophil cell tracking.

5.2 Numerical experiments with real biological data

We present the numerical results illustrating the application of the optimal control algorithm (we refer to Section 4.4) with volume constraint to real biological data from neutrophil migration, as observed in the zebrafish *Danio rerio larvae*. For all the simulations we report in this chapter, for the forward equation (4.9) we set $\varepsilon = 0.1$ and we take the end-time to be $T = 0.05$. For each of the experiments we set the initial guess for the control function to be zero (constant in time and space). For the approximation of the forward and the adjoint equations we use a triangulation with 33025 degrees of freedom and select a uniform time step $\tau = 1 \times 10^{-4}$. Also, we use the same numerical parameters for the optimisation algorithm as for the experiments in chapter 4. They are given in Table 4.2.

We consider the initial data for each experiment to be the biological experimental data as observed at time $t = 0, 1, 2, 3, \ldots, 6$ and the target data to be the neutrophil position at time $t = 2, 3, 4, \ldots, 7$, respectively (Figure 5.1).

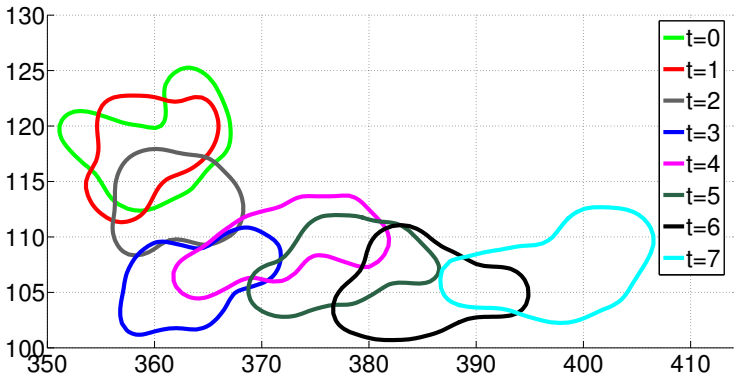
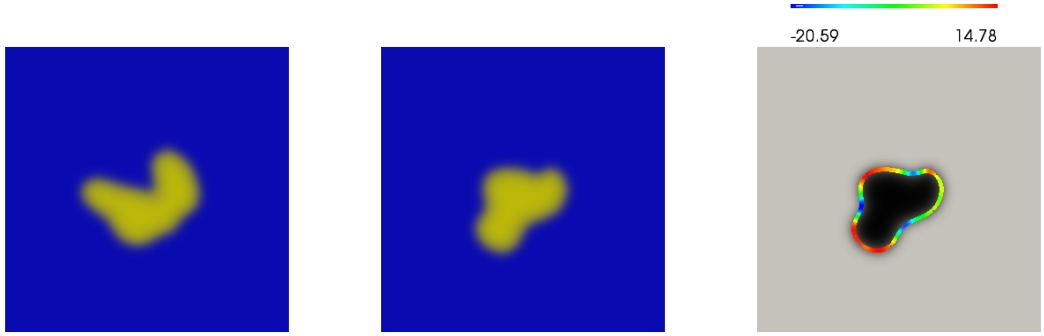


Figure 5.1: Two-dimensional data of migrating zebrafish neutrophils *in vivo* from time $t = 0$ to time $t = 7$ (Henry et al. [2013b]).

To apply our algorithm, based on diffuse interface representations, we defined a mapping from the physical coordinates to the bulk domain $\Omega = [0, 4] \times [0, 4]$. This domain applied to all the experiments contains both the initial and target curves.

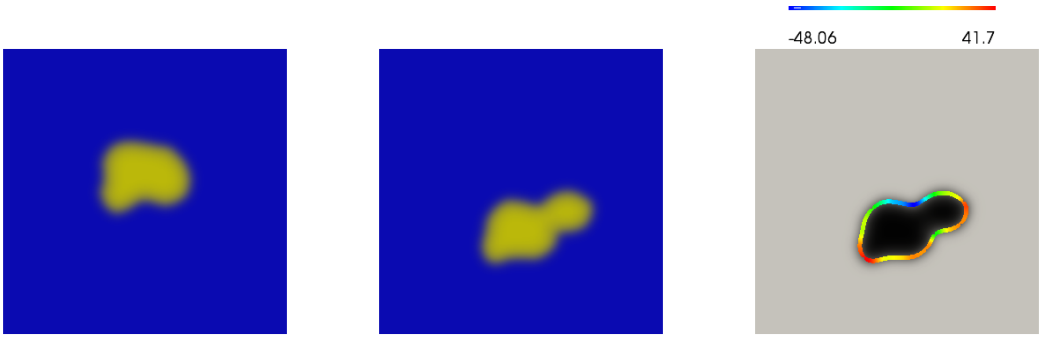
The diffuse interface representation of the initial and target data is constructed according to (4.32). In Figures 5.2 and 5.3 and 5.4, the left and the middle column show the diffuse interface representations of the initial and target data, respectively. The right column in Figures 5.2 and 5.3 and 5.4 presents the zero level-set of the computed solution shaded by the values of the the optimal control at the final time. The background shading corresponds to the target data. In all the experiments the position of the zero level-set of the computed solution shows good agreement with the target data. In addition, we observe for all the experiments, that the computed control is corresponding to strong contractile forces at the "rear" of the neutrophils and to strong protrusive forces at the "front" of the neutrophils (Ananthakrishnan and Ehrlicher [2007]). Figure 5.5 shows the maximum and the minimum values of the control variable $\eta(\boldsymbol{x}, t)$ for the experiment 3 from the neutrophil migration, as observed in zebrafish *Danio rerio larvae*. We observe large increase in the maximum and minimum values as we approach the final time. A possible strategy to prevent this large increase is to impose constraints on the control variable $\eta(\boldsymbol{x}, t)$, as we

noted in Section 4.5.4.



(a) Experiment 1.

(b) Experiment 2.

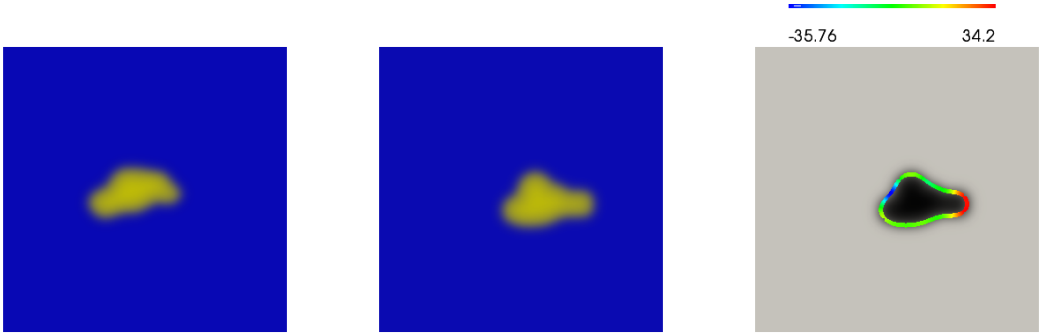


(c) Experiment 3.

Figure 5.2: First and second column: Initial and target data for the experiments with the real biological data from the zebrafish *Danio rerio larvae*. Third column: Zero level-set of the solution using the optimal control. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). We see good agreement between the zero level-set of the data computed with the optimal control and the target data in all the cases.

(a) Experiment 4.

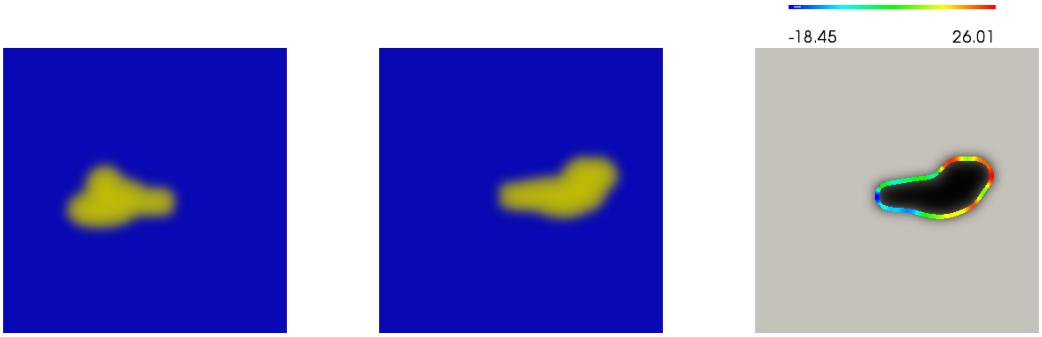
(b) Experiment 5.



(c) Experiment 6.

Figure 5.3: First and second column: Initial and target data for the experiments with the real biological data from the zebrafish *Danio rerio larvae*. Third column: Zero level-set of the solution using the optimal control. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in all the cases.

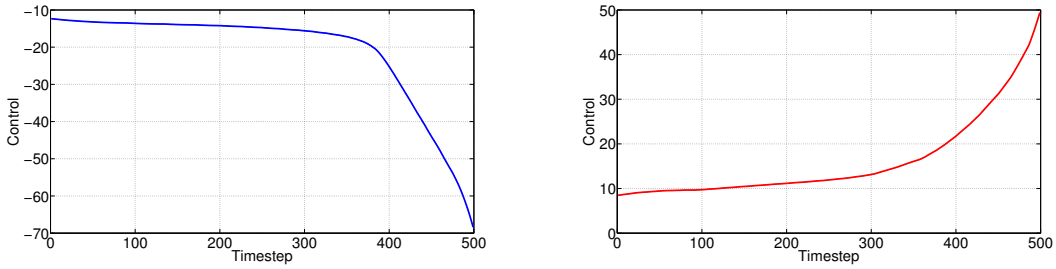
The performance of the algorithm for each experiment is presented in Table5.1.



(a) Experiment 7.

Figure 5.4: First and second column: Initial and target data for the experiments with the real biological data from the zebrafish *Danio rerio larvae*. Third column: Zero level-set of the solution using the optimal control. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in all the cases.

Figure 5.6 shows the plots of the objective functional against the number of iterations of the optimisation algorithm, until it reaches the stopping criteria, for each experiment respectively. The stopping criterion for all the simulations is taken as the change in the solution, through successive time steps, which was less that the predefined tolerance tol_{φ}.



(a) Minimum value of the optimal control. (b) Maximum value of the optimal control.

Figure 5.5: Minimum and maximum values of the control $\eta(\boldsymbol{x}, t)$, for the Experiment3 from the neutrophil migration as observed in zebrafish *Danio rerio larvae*.

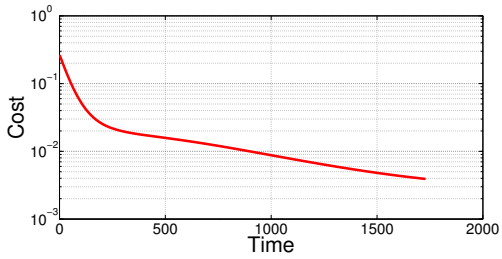
Experiment	Initial time	Final time	Iterations	CPU time(sec)	Figure
1	0	1	1728	426040	5.2(a)
2	1	2	1875	497667	5.2(b)
3	2	3	2153	572398	5.2(c)
4	3	4	2130	565627	5.3(a)
5	4	5	1279	366677	5.3(b)
6	5	6	1033	265849	5.3(c)
7	6	7	2237	594739	5.4(a)

Table 5.1: Performance of the optimal control algorithm with volume constraint until it reaches the stopping criteria for the experiments with the real biological data from the zebrafish *Danio rerio larvae*. The initial and the final time refer to the time t as presented in Figure 5.1.

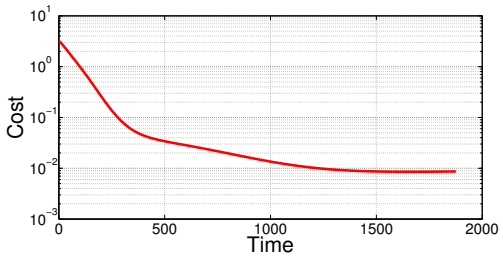
5.3 Comparison between the optimal control algorithm and cubic interpolation

To contrast the optimal control approach with a cell tracking method that is purely geometric in nature (i.e. the trajectories are non-physical), we use the numerical results that we obtained using cubic spline interpolation from Chapter 3. In this section, we will calculate and compare a number of meaningful biological measures that are associated to the study of cell motility, using the data that we extract from both cell tracking algorithms.

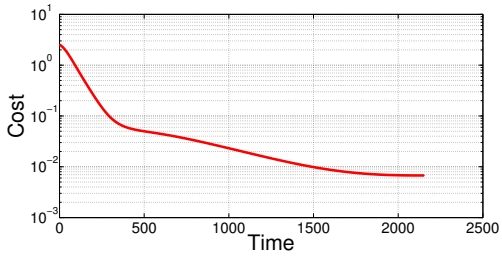
Both cell tracking algorithms, that have been proposed, are able to reconstruct the whole cell morphology from the static image data-set. With the optimal control approach the cell membrane is implicitly defined as the zero level-set of the phase-field variable $\varphi(\boldsymbol{x}, t)$. Whereas with the cubic spline interpolation it is explicitly defined. Therefore, the direct result of applying the tracking methodologies is a sequence of coordinates indicating the locomotion of the cell at every time step. Figures 5.8 and 5.10 show snapshots of the computed neutrophil intermediate positions (blue curve).



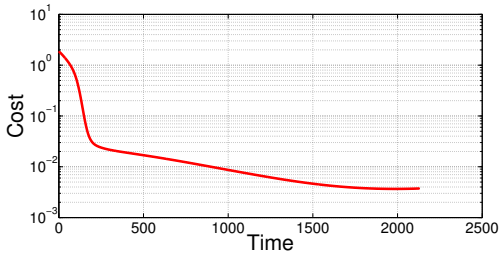
(a) Experiment 1



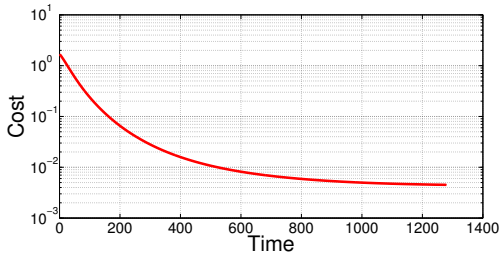
(b) Experiment 2



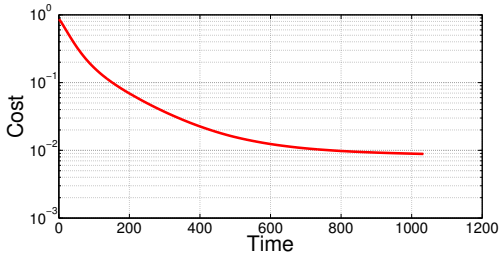
(c) Experiment 3



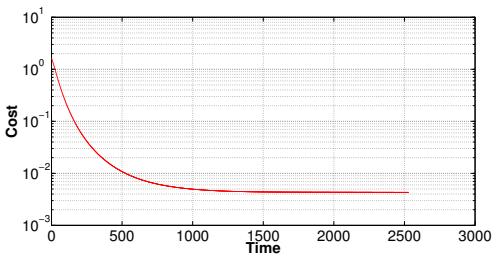
(d) Experiment 4



(e) Experiment 5



(f) Experiment 6



(g) Experiment 7

Figure 5.6: The value of the cost functional versus the number of iterations for the experiments with the real biological data from the zebrafish *Danio rerio larvae*.

In addition, we are able to reconstruct the trajectories of the moving cells. By plotting the trajectory of the cell centroid (centre of mass) over time, it is possible to quantitatively analyse its migratory behaviour. The obtained trajectories can help to identify whether cells show a preference for particular directions (Beltman et al.

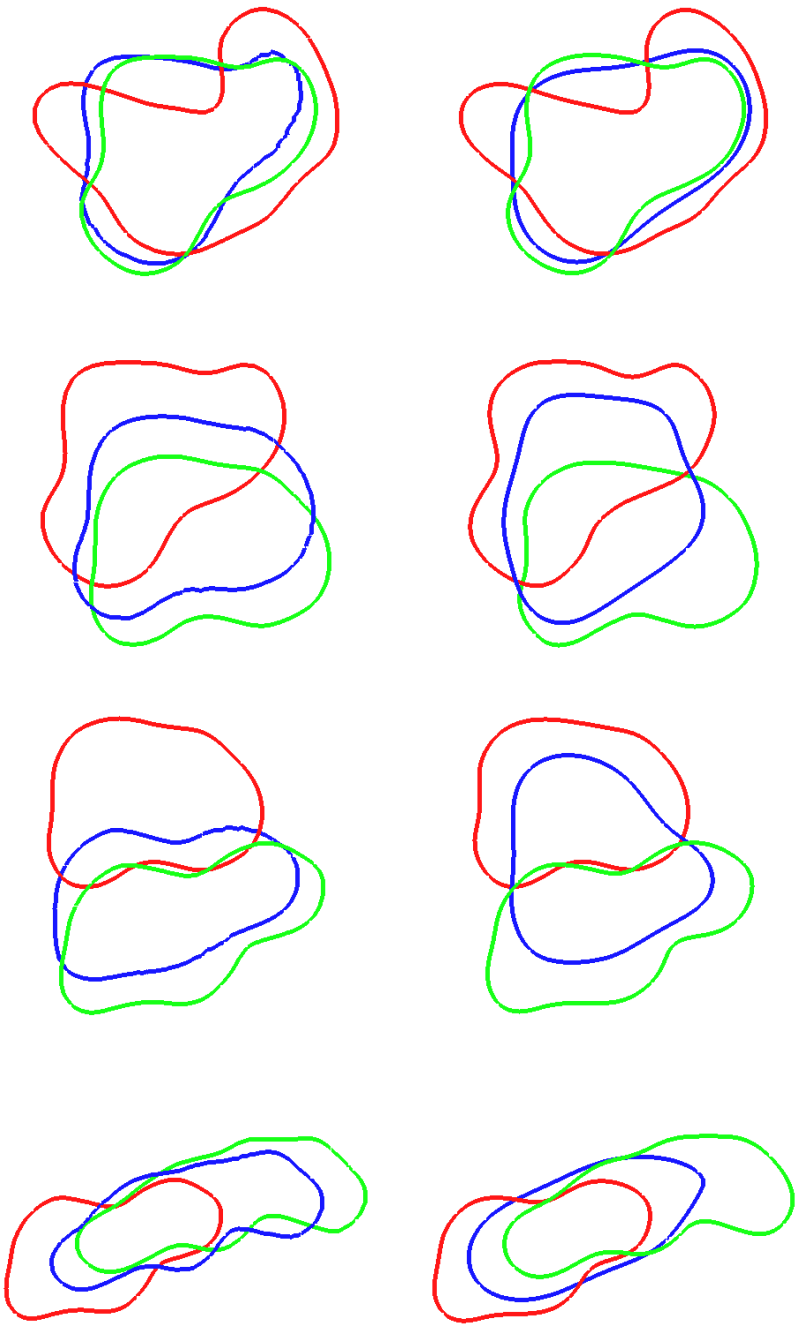


Figure 5.7: First column: Intermediate positions of the locations of the neutrophil after 0 (red), 350 (blue) and 500 (green) time steps, using cubic spline interpolation. Right column: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments with the real biological data from the zebrafish *Danio rerio larvae* after 0 (red), 350 (blue) and 500 (green) time steps.

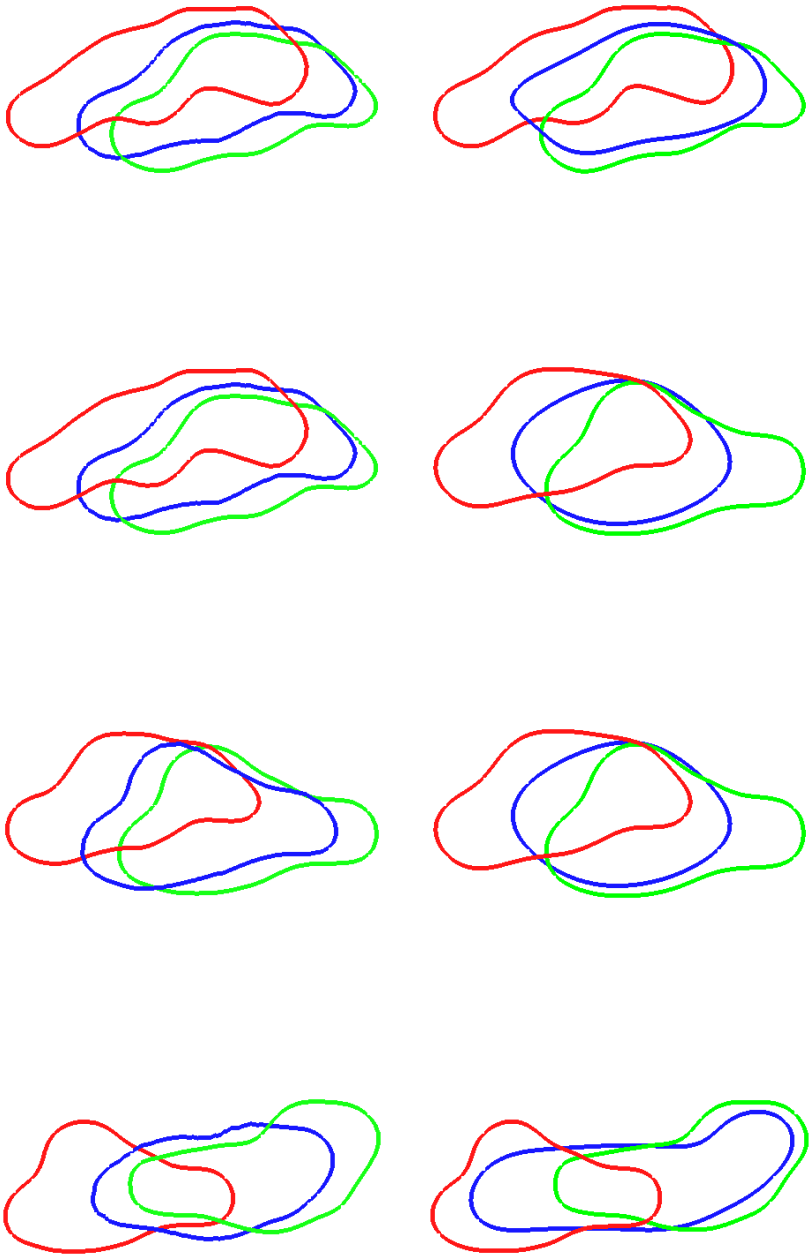
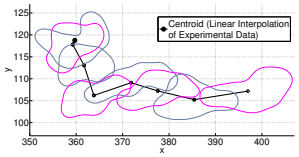


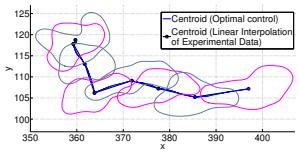
Figure 5.8: First column: Intermediate positions of the locations of the neutrophil after 0 (red), 350 (blue) and 500 (green) time steps, using cubic spline interpolation. Right column: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments with the real biological data from the zebrafish *Danio rerio larvae* after 0 (red), 350 (blue) and 500 (green) time steps.

[2009]; Meijering et al. [2012]).

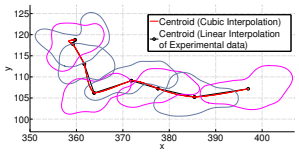
In Figure 5.9 we report on the results of the two tracking algorithms applied to experimental data. We see that both the cubic spline interpolation algorithm and the optimal control algorithm generate centroid trajectories that are significantly smoother than those obtained by linear interpolation of the cell centroids. Figures 5.10 and 5.11 present the neutrophil centroid trajectories for each experiment separately.



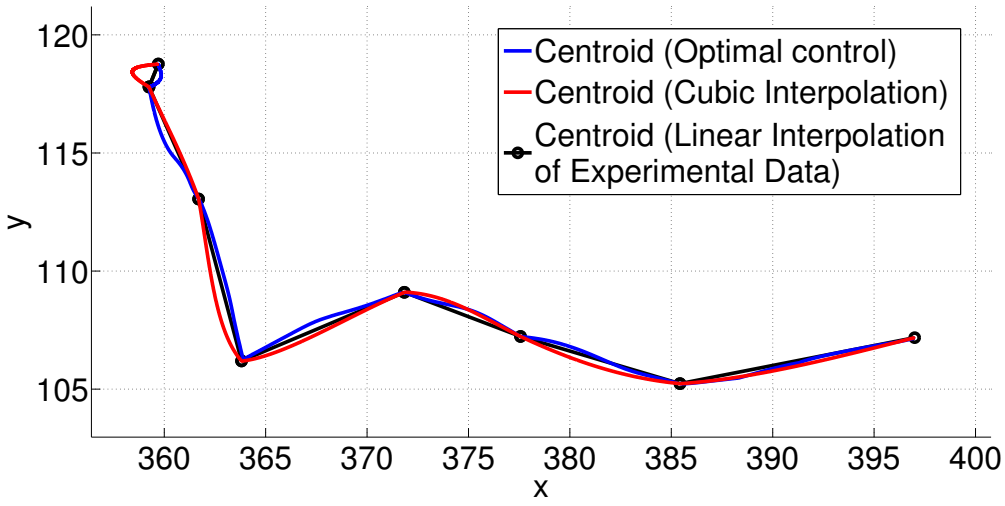
(a) Experimental data



(b) Optimal control approach



(c) Cubic spline interpolation



(d) Compared centroid trajectories

Figure 5.9: (a) Experimental data with the trajectory of their centroid. (b) Centroid trajectories using the optimal control algorithm. (c) Centroid trajectories using the cubic spline interpolation. (d) Compared trajectories of the centroid from linear interpolation of the experiments, using the optimal control algorithm and cubic spline interpolation.

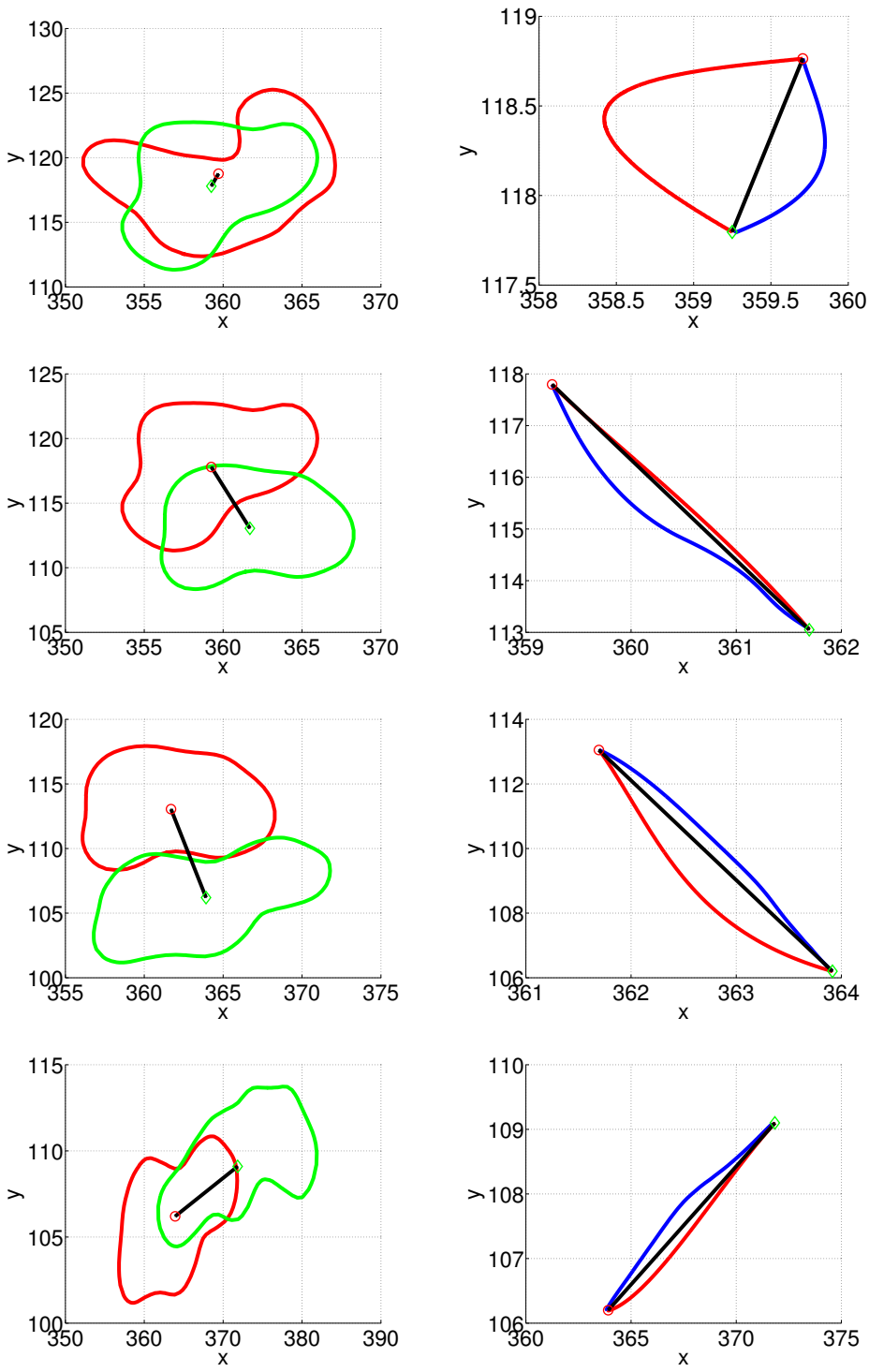


Figure 5.10: Left column: Two-dimensional experimental data of migrating neutrophils from the zebrafish *Danio rerio larvae in vivo*. The red curve denotes the initial and the green the target data. The cell centroids are shown together with the trajectory of the linear interpolant of the cell centroids (black line). Right column: Trajectories of the cell centroids computed by the optimal control algorithm (blue line) and the cubic spline interpolation (red line), respectively. The black line indicates the linear interpolant of the cell centroids.

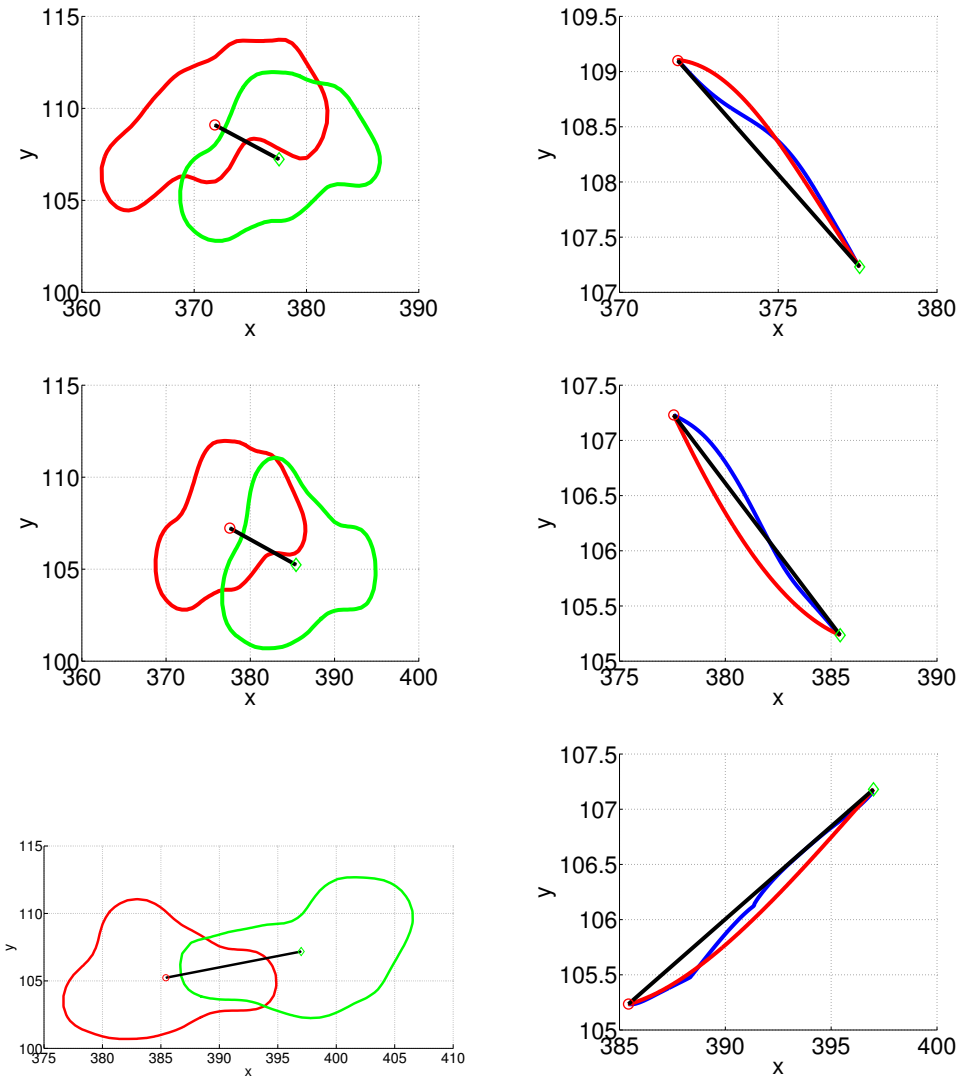
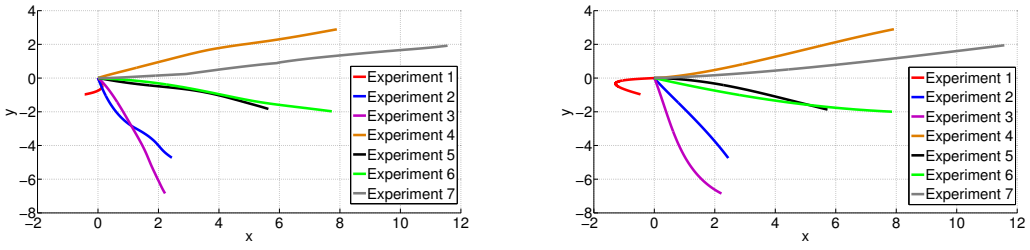


Figure 5.11: Left column: Two-dimensional experimental data of migrating neutrophils from the zebrafish *Danio rerio larvae in vivo*. The red curve denotes the initial and the green the target data. The cell centroids are shown together with the trajectory of the linear interpolant of the cell centroids (black line). Right column: Trajectories of the cell centroids computed by the optimal control algorithm (blue line) and the cubic spline interpolation (red line), respectively. The black line indicates the linear interpolant of the cell centroids.

An alternative measure, which indicates the preferred direction of a moving cell, is to plot the track of each individual cell by starting from the same position and preserve its orientation. If all the possible directions of migration are equally covered, this indicates that the motion is random (Beltman et al. [2009]). The motivation of

the significance of such a measure is a qualitative indication of cell movement. For instance, a chemotactic response should result in the majority of cell trajectories, which start from the same position, to be in the direction of highest chemoattractant concentration. In Figure 5.12 we plot the neutrophil centroid trajectories, by shifting their starting position to the beginning of the axes and preserve their orientation. It is observed that during the three first experiments, the neutrophil prefers a vertical migration. However, during the rest of the experiments, it shows a horizontal migration towards the right side. This finding comes in agreement with the experiments *in vivo*, as the chemoattractant concentration is expected to increase from left to right.



(a) Optimal control approach

(b) Cubic spline interpolation

Figure 5.12: Plot of the neutrophil centroid trajectories, by shifting the starting position to the beginning of the axes and preserve its orientation. We see that during the three first experiments, the neutrophil prefers a vertical migration whilst later prefers more or less a preferred directional migration towards the right side. This finding comes in agreement with the experiments as the chemoattractant concentration is expected to increase from left to right.

Another qualitative measure, which is closely associated with chemotaxis, is the confinement ratio (also known as persistence length) and can be easily extracted from both cell tracking algorithms (Beltman et al. [2009]; Meijering et al. [2012]). It is defined as the ratio of the displacement of the cell to the total length of travelled track, and measures the straightness of the cell's trajectory. The confinement ratio can fluctuate between zero (the cell is returned to the position where it started) and one (the cell trajectory is the straight line between the initial to the final cell position). This occurs because the displacement length is always larger or equal than

the path length. Figure 5.13 presents the confinement ratio for the experiments from the neutrophil migration as observed in zebrafish *Danio rerio larvae*. It is clearly seen that the confinement ratio is close to 1 (apart from the first experiment).

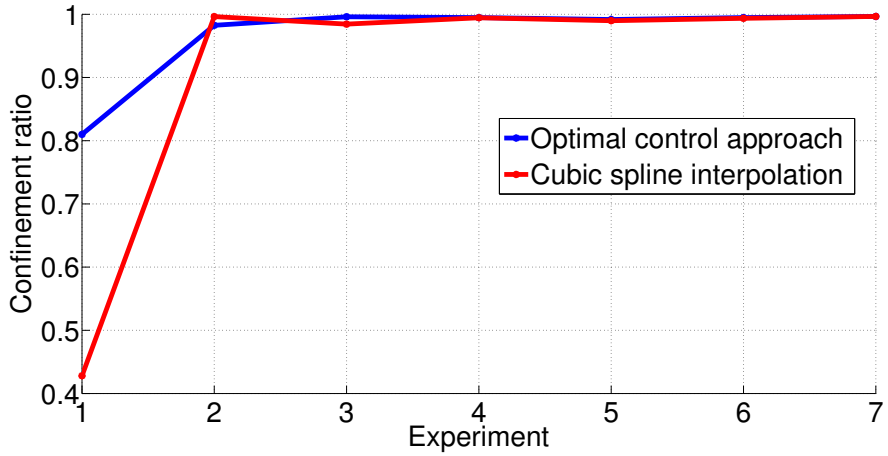


Figure 5.13: Confinement ratio. We observe that for all the experiments (except the first one) that their confinement ratio is close to 1. This indicates that the trajectories are close to a straight line.

A measure, that can be easily derived from the positions of the cell centroids is the speed (which is defined as the magnitude of the velocity). For example, Figure 5.14 shows the speed of the neutrophil's centroid for Experiment 4. We observe an almost constant centroid speed, using the cubic spline interpolation. However, with the optimal control algorithm, a sharp spike in the centroid velocity is observed close to the final time. This increasing centroid speed is unphysical and could be related to the large variation of the maximum and minimum values of the control $\eta(\boldsymbol{x}, t)$. Finally, both proposed algorithms have the capability to compute the area of the tracked cell. In Figure 5.15 we report on the areas of the computed neutrophils. We see that in most of the experiments, with the cubic spline interpolation, the area of the derived neutrophils is close to the linear interpolant. However, using the optimal control algorithm the computed area does not fit well to the linear interpolant. As we noted earlier (we refer to Section 4.5.4), this is a limitation of the proposed algorithm and will be an area for future research and further investigation.

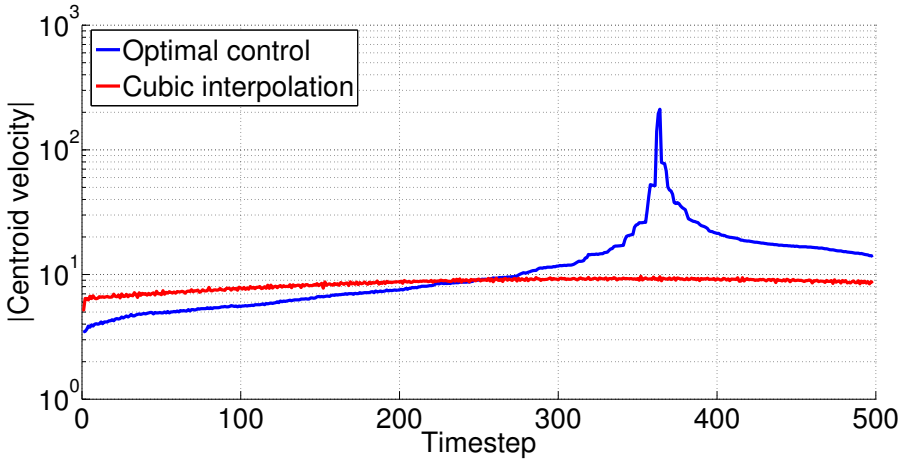
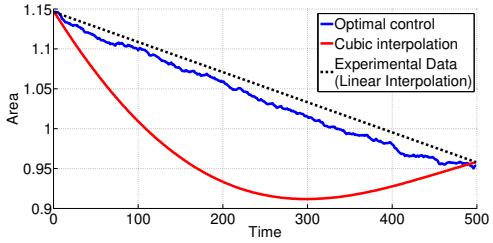


Figure 5.14: Speed of the neutrophil's centroid using the optimal control algorithm and the cubic spline interpolation.

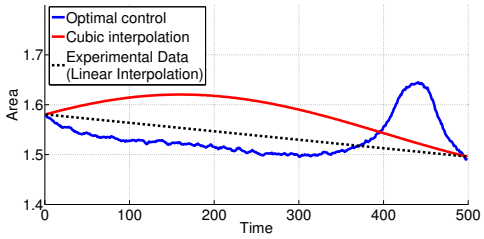
5.4 Conclusion

In this chapter we presented an application of the optimal control algorithm to experimental data on the migration of neutrophils as observed in zebrafish *Danio rerio larvae*. We compared this methodology with the numerical results obtained from Chapter 3, using cubic spline interpolation. Finally, we computed biological meaningful measures related to the study of the cell migration which may be useful in applications. These measurements have been selected to demonstrate the potential of a morphological analysis of neutrophils.

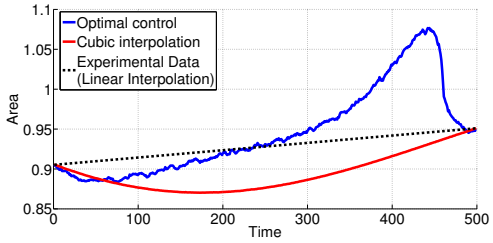
A possible future research could be concentrated in the relationship of these measures and the understanding of neutrophil cell migration. In addition, we believe that the application of optimal control algorithm that we presented in this chapter could be a useful framework within which it could become possible to investigate other biological questions, beyond neutrophil cell tracking, such as the inference of chemotactic field during *in vivo* chemotaxis (Kadirkamanathan et al. [2012]).



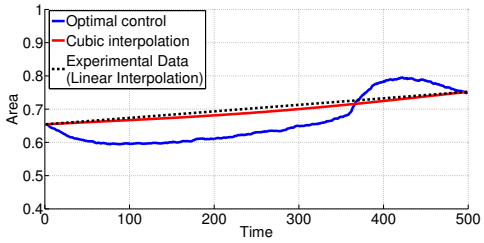
(a) Experiment 1



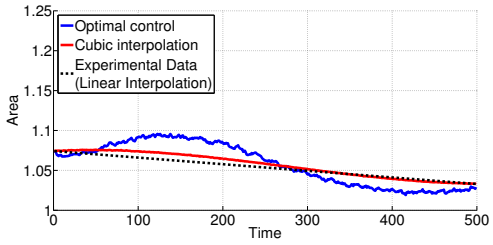
(b) Experiment 2



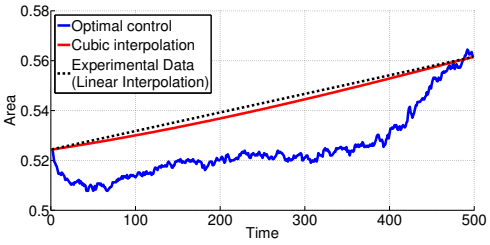
(c) Experiment 3



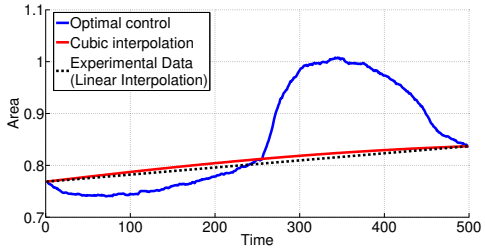
(d) Experiment 4



(e) Experiment 5



(f) Experiment 6



(g) Experiment 7

Figure 5.15: Area of the cell. The black line indicates the linear interpolant of the area of the observations, the blue line the area using the optimal control approach and the red line the area using the cubic spline interpolation.

Chapter 6

Conclusions and future work

6.1 Summary

Within this thesis we consider that "tracking" is equivalent to a dynamic reconstruction of the whole cell data (morphologies) from static image datasets. As in many cell tracking problems, only the position of the cell at a series of discrete times is available, and specifically the cell membrane. No further biological information is given. To this end, we derived a mathematical model for the cell tracking, based on physical principles for cell migration and then formulated an inverse problem. This problem takes the form of a partial differential equation constrained optimisation problem for fitting the model to the experimental observations. We reformulated our model into the phase-field framework. In order to solve the optimisation problem we proposed an algorithm based on previous studies on the optimal control of geometric evolution laws (Haußer et al. [2010]; Haußer et al. [2012]).

The key novelty of this algorithm lies in the fact that the model that drives the tracking procedure (the forward model) is a simplification of recently derived physically motivated models for cell motility. Thus, in contrast to the majority of existing cell tracking algorithms, that consider solely geometric features, our approach is physically motivated and the physics of the model driving the evolution of the cell are reflected in the recovered dynamic data.

In addition, to illustrate the efficiency of our cell tracking algorithm, we presented a number of numerical results for the case of two-dimensional single synthetic and real imaging data, that were selected from the zebrafish *Danio rerio larvae*

neutrophil migration (Henry et al. [2013b]). Also, we demonstrated two examples with multi-cell image data and discussed the capability of our algorithm to solve implicitly the linking problem which is associated in the majority of the cell tracking algorithms. Furthermore, within this thesis we presented a simple cell tracking algorithm based on cubic spline interpolation inspired by Madzvamuse et al. [2003], in which an interpolation method has been applied between the successive given positions of the boundary nodes of a deforming boundary shape. We used our numerical findings to contrast with those that were obtained with the optimal control approach. Using the intermediate cell positions from both algorithms, we computed and discussed biologically meaningful measures which are related to the motility and the morphology of the moving cells (Beltman et al. [2009]; Meijering et al. [2012]).

Finally, in this thesis we have introduced the surface finite element method in order to solve reaction-diffusion equations on the stationary surfaces of a sphere and on those of neutrophils from the zebrafish *Danio rerio larvae* (Dziuk and Elliott [2007b]). As one of the applications of the partial differential equations posed on surfaces is the pattern formation, we conducted a number of computer simulations to solve a non-linear reaction-diffusion system. For illustrative purposes we considered the reaction-diffusion system with Schnakenberg reaction kinetics (Prigogine and Lefever [1968]; Gierer and Meinhardt [1972]; Schnakenberg [1979]). By taking a fixed set of parameters values and changing the surface geometry, we observed the emergence of different patterns.

Moreover, surface partial differential equations have been used to develop robust computational models for investigating cell motility (Neilson et al. [2010]; Ziebert et al. [2011]; Neilson et al. [2011a]; Neilson et al. [2011b]; Elliott et al. [2012]; Shao et al. [2012]; Marth and Voigt [2013]; Croft et al. [2013]). Regularly, such equations are associated to describe the biochemistry of the models. Although in this thesis we have focused on the case where no information regarding the biochemistry of the examined cell is available, our proposed optimal control algorithm is applicable and generalises to models where more biological aspects are included. In particular models for the dynamics of actin and myosin, as well as other motility related species within the cell and on the membrane may be included, and the evolution law may be modified to take into account the dependence of the movement of the cell on

these species.

6.2 Future work

We can now underline possible directions for the improvement of the current theme study and future research:

- Like other algorithms, the proposed cell tracking algorithm which is based on the phase-field and optimal control approach, has limitations. A future step for the improvement of this algorithm is to impose pointwise constraints on the control function $\eta(\boldsymbol{x}, t)$.

- The model we used, which drives the optimal control cell tracking algorithm, is phenomenological and is difficult to directly relate η to biophysical processes. However, our numerical findings indicate that as positive values of η correspond to protrusive forces and negative values of η correspond to contractile forces, one interpretation of the forcing function η is that it accounts for both protrusive forces generated by polymerisation of actin at the leading edge of the cell together with contractile forces generated by the action of myosin motors at the cell rear. Thus, a potential avenue for assessing the plausibility of the cell tracks computed with our algorithm would be to compare the computed η with experimental imaging data on the location of polymerised actin and myosin-II on the cell membrane with the expectation being that regions in which the computed forcing η is positive would correspond to regions rich in polymerised actin and regions in which the computed forcing η is negative would correspond to regions rich in myosin-II.

- In addition, the diffuse interface or phase-field framework that we employed does make the proposed algorithm computationally intensive as evidenced by the relatively large CPU times for our experiments. A key area for future work is to investigate improvements in the computational efficiency of the algorithm. This need is especially evident if one wishes to track cells in $3d$, as although our theoretical framework applies equally to this setting the computational cost becomes prohibitive.

- Spatial and temporal adaptivity which is challenging in this setting as the solution of the state equation enters the adjoint equation (Haußer et al. [2010]; Haußer et al. [2012]).

- We could consider the second order optimisation Newton's method (e.g. Snyman [2005]) as an alternative update scheme for the control variable $\eta(\boldsymbol{x}, t)$ (4.29).

- Parallelisation and the development of fast solvers for the solution of the state and adjoint equations.

- The model we proposed for the evolution of the cell in this thesis is a simplification of more general physically relevant models in which bulk or surface partial differential equations for the biochemistry are coupled to a geometric evolution law for the motion. An important area for future work is the extension of the framework to this more general setting.

- Finally, we believe that the optimal control approach we presented in this thesis could be a useful framework with which it could become possible to investigate other biological questions, beyond cell tracking such as the inference of chemotactic fields during *in vivo* chemotaxis (Kadirkamanathan et al. [2012]).

Bibliography

Acheson, D. (1990). *Elementary fluid dynamics*. Oxford University Press, USA.

Adams, R. and Fournier, J. (2003). Sobolev spaces, volume 140 of Pure and Applied Mathematics.

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., and Watson, J. D. (1994). Molecular biology of the cell. *Garland, New York*, pages 139–194.

Ananthakrishnan, R. and Ehrlicher, A. (2007). The forces behind cell movement. *International journal of biological sciences*, 3(5):303.

Anderson, D., McFadden, G. B., and Wheeler, A. (1998). Diffuse-interface methods in fluid mechanics. *Annual review of fluid mechanics*, 30(1):139–165.

Andreoli, T. E., Benjamin, I., Griggs, R. C., Wing, E. J., and Fitz, J. G. (2010). *Andreoli and Carpenter's Cecil essentials of medicine*. Elsevier Health Sciences.

Ascher, U. M., Ruuth, S. J., and Wetton, B. T. (1995). Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823.

Barreira, R., Elliott, C. M., and Madzvamuse, A. (2011). The surface finite element method for pattern formation on evolving biological surfaces. *Journal of mathematical biology*, 63(6):1095–1119.

Bellettini, G. and Paolini, M. (1996). Anisotropic motion by mean curvature in the context of finsler geometry. *Hokkaido Mathematical Journal*, 25(3):537–566.

Beltman, J. B., Marée, A. F., and de Boer, R. J. (2009). Analysing immune cell migration. *Nature Reviews Immunology*, 9(11):789–798.

Bergdorf, M., Sbalzarini, I. F., and Koumoutsakos, P. (2010). A lagrangian particle method for reaction–diffusion systems on deforming surfaces. *Journal of mathematical biology*, 61(5):649–663.

Blank, L., Farshbaf-Shaker, H., Garcke, H., and Styles, V. (2013). Relating phase field and sharp interface approaches to structural topology optimization.

Blazakis, K., Reyes-Aldasoro, C.-C., Styles, V., Venkataraman, C., and Madzvamuse, A., editors (2015a). *An optimal control approach to cell tracking*. In: Louis, Alfred K, Arridge, Simon and Rundell, Bill (eds.), Proceedings of the Inverse Problems from Theory to Applications Conference (IPTA2014). IOP Publishing Ltd, pp. 74-77.

Blazakis, K. N., Madzvamuse, A., Reyes-Aldasoro, C.-C., Styles, V., and Venkataraman, C. (2015b). Whole cell tracking through the optimal control of geometric evolution laws. *Journal of Computational Physics*, 297:495–514.

Blowey, J. and Elliott, C. (1993). Curvature dependent phase boundary motion and parabolic double obstacle problems. In *Degenerate Diffusions*, pages 19–60. Springer.

Bosgraaf, L., van Haastert, P., and Bretschneider, T. (2009). Analysis of cell movement by simultaneous quantification of local membrane displacement and fluorescent intensities using quimp2. *Cell motility and the cytoskeleton*, 66(3):156–165.

Bosgraaf, L. and Van Haastert, P. J. (2010). Quimp3, an automated pseudopod-tracking algorithm. *Cell adhesion & migration*, 4(1):46.

Brassel, M. and Bretin, E. (2011). A modified phase field approximation for mean curvature flow with conservation of the volume. *Mathematical Methods in the Applied Sciences*, 34(10):1157–1180.

Bray, D. (2001). *Cell movements: from molecules to motility*. Routledge.

Burger, M. (2009). Finite element approximation of elliptic partial differential equations on implicit surfaces. *Computing and visualization in science*, 12(3):87–100.

Chan, T. F. and Vese, L. A. (2001). Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277.

Chen, X. (1992). Generation and propagation of interfaces in reaction-diffusion systems. *Transactions of the American Mathematical Society*, 334(2):877–913.

Cheng, S., Heilman, S., Wasserman, M., Archer, S., Shuler, M., and Wu, M. (2007). A hydrogel-based microfluidic device for the studies of directed cell migration. *Lab Chip*, 7(6):763–769.

Crampin, E. J., Gaffney, E. A., and Maini, P. K. (1999). Reaction and diffusion on growing domains: scenarios for robust pattern formation. *Bulletin of mathematical biology*, 61(6):1093–1120.

Croft, W., Elliott, C., Ladds, G., Stinner, B., Venkataraman, C., and Weston, C. (2013). Parameter identification problems in the modelling of cell motility. *arXiv preprint arXiv:1311.7602*.

De Boor, C. (1978). *A practical guide to splines*. Springer-Verlag, New York.

Deckelnick, K., Dziuk, G., and Elliott, C. (2005). Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232.

Deckelnick, K., Elliott, C. M., and Styles, V. (2001). Numerical diffusion-induced grain boundary motion. *Interfaces and Free Boundaries*, 3(4):393–414.

Dormann, D., Libotte, T., Weijer, C. J., and Bretschneider, T. (2002). Simultaneous quantification of cell motility and protein-membrane-association using active contours. *Cell motility and the cytoskeleton*, 52(4):221–230.

Du, Q., Liu, C., Ryham, R., and Wang, X. (2005). Modeling the spontaneous curvature effects in static cell membrane deformations by a phase field formulation. *Energy*, 7:8.

Du, Q., Liu, C., and Wang, X. (2006). Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions. *Journal of Computational Physics*, 212(2):757–777.

Dziuk, G. (1988). *Finite elements for the Beltrami operator on arbitrary surfaces*. Springer.

Dziuk, G. and Elliott, C. M. (2007a). Finite elements on evolving surfaces. *IMA journal of numerical analysis*, 27(2):262–292.

Dziuk, G. and Elliott, C. M. (2007b). Surface finite elements for parabolic equations. *Journal of Computational Mathematics-International Editions*, 25(4):385.

Dziuk, G. and Elliott, C. M. (2008). Eulerian finite element method for parabolic pdes on implicit surfaces. *Interfaces and Free Boundaries*, 10(119-138):464.

Dziuk, G. and Elliott, C. M. (2010). An eulerian approach to transport and diffusion on evolving implicit surfaces. *Computing and visualization in science*, 13(1):17–28.

Dziuk, G. and Elliott, C. M. (2012). A fully discrete evolving surface finite element method. *SIAM Journal on Numerical Analysis*, 50(5):2677–2694.

Dziuk, G. and Elliott, C. M. (2013). Finite element methods for surface PDEs. *Acta Numerica*, 22:289–396.

Edwards, S. W. (2005). *Biochemistry and Physiology of the Neutrophil*. Cambridge University Press.

Elks, P. M., Loynes, C. A., and Renshaw, S. A. (2011). Measuring inflammatory cell migration in the zebrafish. In Wells, C. M. and Parsons, M., editors, *Cell Migration*, pages 261–275. Springer.

Elliott, C. M. and Ranner, T. (2012). Finite element analysis for a coupled bulk–surface partial differential equation. *IMA Journal of Numerical Analysis*.

Elliott, C. M., Stinner, B., Styles, V., and Welford, R. (2010). Numerical computation of advection and diffusion on evolving diffuse interfaces. *IMA Journal of Numerical Analysis*.

Elliott, C. M., Stinner, B., and Venkataraman, C. (2012). Modelling cell motility and chemotaxis with evolving surface finite elements. *Journal of The Royal Society Interface*.

Evans, L. (2009). *Partial Differential Equations (Graduate Studies in Mathematics, Vol. 19)*. Dover.

Evans, L. C., Soner, H. M., and Souganidis, P. E. (1992). Phase transitions and generalized motion by mean curvature. *Communications on Pure and Applied Mathematics*, 45(9):1097–1123.

Fedkiw, S. O. R. (2003). Level set methods and dynamic implicit surfaces.

Flaherty, B., McGarry, J., and McHugh, P. (2007). Mathematical models of cell motility. *Cell biochemistry and biophysics*, 49(1):14–28.

Fletcher, D. A. and Theriot, J. A. (2004). An introduction to cell motility for the physical scientist. *Physical Biology*, 1(1):T1.

Ganguly, S., Dora, K., Sarkar, S., Chowdhury, S., and Ahmed, S. (2012). Zebra fish-general biology and a biomedical research tool. *International Journal of Science, Environment and Technology*, 1(3):193–196.

Gierer, A. and Meinhardt, H. (1972). A theory of biological pattern formation. *Kybernetik*, 12(1):30–39.

Gilbarg, D. and Trudinger, N. (1977). Elliptic partial differential equations of second order,. *Springer-Verlag Berlin Heidelberg New York*, 401:58.

Hagan, P. S. and West, G. (2006). Interpolation methods for curve construction. *Applied Mathematical Finance*, 13(2):89–129.

Hand, A., Sun, T., Barber, D., Hose, D., and MacNeil, S. (2009). Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration. *Journal of microscopy*, 234(1):62–79.

Haußer, F., Janssen, S., and Voigt, A. (2012). Control of nanostructures through electric fields and related free boundary problems. In Leugering, G., editor, *Constrained Optimization and Optimal Control for Partial Differential Equations*, pages 561–572. Springer Basel.

Haußer, F., Rasche, S., and Voigt, A. (2010). The influence of electric fields on nanostructures—simulation and control. *Mathematics and Computers in Simulation*, 80(7):1449–1457.

Henderson, A., Ahrens, J., and Law, C. (2004). *The ParaView Guide*. Kitware Clifton Park, NY.

Henry, K. M., Loynes, C. A., Whyte, M. K., and Renshaw, S. A. (2013a). Zebrafish as a model for the study of neutrophil biology. *Journal of leukocyte biology*, 94(4):633–642.

Henry, K. M., Pase, L., Ramos-Lopez, C. F., Lieschke, G. J., Renshaw, S. A., and Reyes-Aldasoro, C. C. (2013b). Phagosight: an open-source matlab® package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model. *PloS one*, 8(8):e72636.

Heys, J., DeGroff, C., Manteuffel, T., and McCormick, S. (2006). First-order system least-squares (fosls) for modeling blood flow. *Medical engineering & physics*, 28(6):495–503.

Hieber, S. E. and Koumoutsakos, P. (2005). A lagrangian particle level set method. *Journal of Computational Physics*, 210(1):342–367.

Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. (2009). Optimization with pde constraints. *Mathematical Modelling: Theory and Applications.*, 23.

Holmes, G. R., Anderson, S. R., Dixon, G., Robertson, A. L., Reyes-Aldasoro, C. C., Billings, S. A., Renshaw, S. A., and Kadirkamanathan, V. (2012). Repelled from the wound, or randomly dispersed? reverse migration behaviour of neutrophils characterized by dynamic modelling. *Journal of The Royal Society Interface*, 9(77):3229–3239.

Horwitz, R. and Webb, D. (2003). Cell migration. *Current Biology*, 13(19):R756–R759.

Iwashita, Y. (2013). Piecewise polynomial interpolations. , Open Gamma Technical Report.

Kadirkamanathan, V., Anderson, S. R., Billings, S. A., Zhang, X., Holmes, G. R., Reyes-Aldasoro, C. C., Elks, P. M., and Renshaw, S. A. (2012). The neutrophil's eye-view: Inference and visualisation of the chemoattractant field driving cell chemotaxis in vivo. *PloS one*, 7(4):e35182.

Kreyszig, E. (2007). *Advanced engineering mathematics*. John Wiley & Sons.

Landsberg, C. and Voigt, A. (2010). A multigrid finite element method for reaction-diffusion systems on surfaces. *Computing and Visualization in Science*, 13(4):177–185.

Le Clainche, C. and Carlier, M.-F. (2008). Regulation of actin assembly associated with protrusion and adhesion in cell migration. *Physiological reviews*, 88(2):489–513.

Lefèvre, J. and Mangin, J. (2010). A reaction-diffusion model of human brain development. *PLoS computational biology*, 6(4):e1000749.

Lehmann, T. M., Gonner, C., and Spitzer, K. (1999). Survey: Interpolation methods in medical image processing. *Medical Imaging, IEEE Transactions on*, 18(11):1049–1075.

Lieschke, G. J. and Currie, P. D. (2007). Animal models of human disease: zebrafish swim into view. *Nature Reviews Genetics*, 8(5):353–367.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM.

Lubich, C., Mansour, D., and Venkataraman, C. (2013). Backward difference time discretization of parabolic differential equations on evolving surfaces. *IMA Journal of Numerical Analysis*, 33(4):1365–1385.

Macdonald, C. B., Merriman, B., and Ruuth, S. J. (2013). Simple computation of reaction–diffusion processes on point clouds. *Proceedings of the National Academy of Sciences*, 110(23):9209–9214.

Madzvamuse, A. (2000). *A numerical approach to the study of spatial pattern formation.* PhD thesis, University of Oxford.

Madzvamuse, A. (2006). Time-stepping schemes for moving grid finite elements applied to reaction–diffusion systems on fixed and growing domains. *Journal of computational physics*, 214(1):239–263.

Madzvamuse, A. (2007). A modified backward Euler scheme for advection-reaction-diffusion systems. *Mathematical Modeling of Biological Systems, Volume I*, pages 183–189.

Madzvamuse, A. and Chung, A. H. (2014). Fully implicit time-stepping schemes and non-linear solvers for systems of reaction–diffusion equations. *Applied Mathematics and Computation*, 244:361–374.

Madzvamuse, A., Wathen, A., and Maini, P. (2003). A moving grid finite element method applied to a model biological pattern generator. *J Comput Phys*, 190(2):478–500.

Maini, P. K., Painter, K. J., and Chau, H. N. P. (1997). Spatial pattern formation in chemical and biological systems. *Journal of the Chemical Society, Faraday Transactions*, 93(20):3601–3610.

Marchet, C. (2012). Biomedical engineering: Characterisation of three-dimensional modeling of neutrophils in vivo. Research Intership, University of Sussex.

Marth, W. and Voigt, A. (2013). Signaling networks and cell motility: a computational approach using a phase field description. *Journal of mathematical biology*, pages 1–22.

Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D. M., et al. (2014). A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617.

Meijering, E., Dzyubachyk, O., Smal, I., et al. (2012). Methods for cell and particle tracking. *Methods Enzymol*, 504(9):183–200.

Meijering, E., Dzyubachyk, O., Smal, I., and van Cappellen, W. A. (2009). Tracking in cell and developmental biology. *Seminars in Cell Developmental Biology*, 20(8):894–902.

Miklos, P. (2004). Image interpolation techniques. In *2nd Siberian-Hungarian Joint Symposium On Intelligent Systems*.

Miura, K. (2005). Tracking movement in cell biology. In Rietdorf, J., editor, *Microscopy Techniques*, pages 267–295. Springer.

Murray, J. D. (2002). *Mathematical Biology I: An Introduction, vol. 17 of Interdisciplinary Applied Mathematics*. Springer, New York, NY, USA,.

Najia, X., Xi, L., and Lixin, L. (2011). Tracking neutrophil intraluminal crawling, transendothelial migration and chemotaxis in tissue by intravital video microscopy. *Journal of Visualized Experiments*, (55).

Nathan, C. (2006). Neutrophils and immunity: challenges and opportunities. *Nat. Rev. Immunol.*, 6(3):173–182.

Neilson, M., Mackenzie, J., Webb, S., and Insall, R. (2011a). Modelling cell movement and chemotaxis pseudopod based feedback. *SIAM Journal on Scientific Computing*, 33(3):1035–1057.

Neilson, M., Veltman, D., van Haastert, P., Webb, S., Mackenzie, J., and Insall, R. (2011b). Chemotaxis: A feedback-based computational model robustly predicts multiple aspects of real cell behaviour. *PLoS biology*, 9(5):e1000618.

Neilson, M. P., Mackenzie, J. A., Webb, S. D., and Insall, R. H. (2010). Use of the parameterised finite element method to robustly and efficiently evolve the edge of a moving cell. *Integrative Biology*, 2(11-12):687–695.

Pittet, M. J. and Weissleder, R. (2011). Intravital imaging. *Cell*, 147(5):983–991.

Plaza, R. G., Sanchez-Garduno, F., Padilla, P., Barrio, R. A., and Maini, P. K. (2004). The effect of growth and curvature on pattern formation. *Journal of Dynamics and Differential Equations*, 16(4):1093–1121.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1990). *Numerical recipes*. Cambridge University Press.

Prigogine, I. and Lefever, R. (1968). Symmetry breaking instabilities in dissipative systems. ii. *The Journal of Chemical Physics*, 48(4):1695–1700.

Rätz, A. and Voigt, A. (2006). Pde's on surfaces—a diffuse interface approach. *Communications in Mathematical Sciences*, 4(3):575–590.

Renshaw, S. A., Loynes, C. A., Trushell, D. M., Elworthy, S., Ingham, P. W., and Whyte, M. K. (2006). A transgenic zebrafish model of neutrophilic inflammation. *Blood*, 108(13):3976–3978.

Reyes-Aldasoro, C., Akerman, S., and Tozer, G. (2008). Measuring the velocity of fluorescently labelled red blood cells with a keyhole tracking algorithm. *Journal of microscopy*, 229(1):162–173.

Reyes-Aldasoro, C. C., Zhao, Y., Coca, D., Billings, S. A., Kadirkamanathan, V., Tozer, G. M., and Renshaw, S. A. (2009). Analysis of immune cell function using in vivo cell shape analysis and tracking. In *4th IAPR Int Conf Patt Recog Bioinformatics*, pages 7–9.

Ridley, A. J., Schwartz, M. A., Burridge, K., Firtel, R. A., Ginsberg, M. H., Borisy, G., Parsons, J. T., and Horwitz, A. R. (2003). Cell migration: integrating signals from front to back. *Science*, 302(5651):1704–1709.

Ruuth, S. and Merriman, B. (2008). A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3):1943–1961.

Ruuth, S. J. (1995). Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34(2):148–176.

Schmidt, A. and Siebert, K. (2005). *Design of adaptive finite element software: The finite element toolbox ALBERTA*. Springer Verlag.

Schnakenberg, J. (1979). Simple chemical reaction systems with limit cycle behaviour. *Journal of theoretical biology*, 81(3):389.

Sethian, J. (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Number 3. Cambridge Univ Press.

Shao, D., Levine, H., and Rappel, W.-J. (2012). Coupling actin flow, adhesion, and morphology in a computational cell motility model. *Proceedings of the National Academy of Sciences*, 109(18):6851–6856.

Shao, D., Rappel, W., and Levine, H. (2010). Computational model for cell morphodynamics. *Physical review letters*, 105(10):108104.

Snyman, J. (2005). *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, volume 97. Springer.

Spence, R., Gerlach, G., Lawrence, C., and Smith, C. (2008). The behaviour and ecology of the zebrafish, danio rerio. *Biological Reviews*, 83(1):13–34.

Steinbach, I. (2009). Phase-field models in materials science. *Modelling and Simulation in Materials Science and Engineering*, 17(7):073001.

Tang, P., Qiu, F., Zhang, H., and Yang, Y. (2005). Phase separation patterns for diblock copolymers on spherical surfaces: A finite volume method. *Physical Review E*, 72(1):016710.

Tröltzsch, F. (2010). *Optimal control of partial differential equations: Theory, methods and applications*. AMS Bookstore.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.

Turk, G. (1991). *Generating textures on arbitrary surfaces using reaction-diffusion*. Cumputer Graphics (SIGGRAPH '91 Proceedings) Vol. 25, 289-298.

Tyson, R. A., Epstein, D., Anderson, K., and Bretschneider, T. (2010). High resolution tracking of cell membrane dynamics in moving cells: an electrifying approach. *Mathematical Modelling of Natural Phenomena*, 5(01):34–55.

Varea, C., Aragon, J., and Barrio, R. (1999). Turing patterns on a sphere. *Physical Review E*, 60(4):4588.

Venkataraman, C., Sekimura, T., Gaffney, E. A., Maini, P. K., and Madzvamuse, A. (2011). Modeling parr-mark pattern formation during the early development of amago trout. *Physical Review E*, 84(4):041923.

Wang, X. (2008). Method of steepest descent and its applications. *IEEE Microwave and Wireless Components Letters*, 12:24–26.

Wolf, K., Mazo, I., Leung, H., Engelke, K., Von Andrian, U. H., Deryugina, E. I., Strongin, A. Y., Bröcker, E.-B., and Friedl, P. (2003). Compensation mechanism in tumor cell migration mesenchymal–amoeboid transition after blocking of pericellular proteolysis. *The Journal of cell biology*, 160(2):267–277.

Xiong, Y. and Iglesias, P. A. (2010). Tools for analyzing cell shape changes during chemotaxis. *Integrative Biology*, 2(11-12):561–567.

Xue, F., Janzen, D. M., and Knecht, D. A. (2010). Contribution of filopodia to cell migration: a mechanical link between protrusion and contraction. *International journal of cell biology*, 2010.

Yap, B. and Kamm, R. D. (2005). Mechanical deformation of neutrophils into narrow channels induces pseudopod projection and changes in biomechanical properties. *Journal of Applied Physiology*, 98(5):1930–1939.

Ziebert, F., Swaminathan, S., and Aranson, I. (2011). Model for self-polarization and motility of keratocyte fragments. *Journal of The Royal Society Interface*.

Zimmer, C., Zhang, B., Dufour, A., Thébaud, A., Berlemont, S., Meas-Yedid, V., and Marin, J.-C. (2006). On the digital trail of mobile cells. *Signal Processing Magazine, IEEE*, 23(3):54–62.